THALES

cpl.thalesgroup.com

payShield 9000 v3.5

Host Command Reference Manual

Addendum for Licenses LIC011, LIC016, LIC018 & LIC023

(Card & Mobile Issuance Commands)

1270A548-038

26 July 2021

Contents

End User License Agreement

Use of this product is subject to the Thales Cloud Protection & Licensing *End User License Agreement* found at:

https://cpl.thalesgroup.com/legal

Revision Status

Document No.	Manual Set	Software Version	Release Date
1270A548-038	Issue 38	payShield 9000 v3.5	July 2021

References

The following documents are referenced in this document:

1	payShield 9000 HSM Host Command Reference Manual Document Number 1270A546, Issue 1
2	PKCS#1: RSA Encryption Standard - Version 1.5 – Revised November 1993 (www.rsalabs.com)
3	PKCS#1: RSA Cryptography Standard – Version 2.0 – October 1998 (www.rsalabs.com)
4	EMV '96 Integrated Circuit Card Specification for Payment Systems, Version 3.1.1, May 1998
5	EMV Integrated Circuit Card Specification for Payment Systems, Book 2 – Security and Key Management Version 4.1, May 2004
6	EMV Specification Update Bulletin No. 46 First Edition October 2005
7	American Express – Global Network Services AEIPS Hardware Security Module (HSM) Specification (AEIPS4.1), June 2005
8	Mastercard PayPass – Mag Stripe Technical Specifications Version 3.2, December 2007
9	Mastercard PayPass – Mag Stripe Application Note #12 Errata – January 27, 2007
10	Mastercard M/Chip 4 Version 1.1 Security and Key Management, June 2006
11	Visa Integrated Circuit Card – Card Specification Version 1.4.0 – October 31, 2001 including corrections dated November 1, 2001
12	Visa Integrated Circuit Card – Application Overview Version 1.4.0 – October 31, 2001 including corrections dated November 1, 2001
13	Visa Contactless Payment Specification Version 2.0.2, July 2006
14	Visa Smart Debit/Credit (VSDC) – Technical Guide to Visa's Applet For GlobalPlatform Cards, March 2007
15	MULTOS – Guide to Generating Application Load Units Document # MAO-DOC-TEC-009, Version 2.52 (www.multos.com)
16	MULTOS step/one Off-Card Specification Document # MAO-ONE-OFF-0001, Version 1, May 2004
17	EMV Card Personalization Specification Version 1.1, July 2007 (www.emvco.com)
18	Global Platform – Guide to Common Personalization Version 1.0, March 2003 (www.globalplatform.org)
19	Global Platform – Card Specification Version 2.2, March 2006 (www.globalplatform.org)
20	Discover® Network RF Contactless Specification – Supplement Guide for Functional Portion, Version 2.12, August 2009

Chapter 1 – Introduction

Overview

The purpose of this addendum is to define the separately licensed HSM Command Set to perform the functions required for Contactless and EMV-based card issuing.

PCI HSM Certification and Compliance

See Chapter 10 of the *payShield 9000 General Information Manual* for information about PCI HSM certification of the payShield 9000.

Key Type Table

See the Key Type Table in Chapter 7 of the *payShield 9000 Host Programmer's Manual*.

Key Scheme Table

Whenever a key is entered into the HSM, it must be prefixed by a Key Scheme Tag which allows the HSM to interpret the key correctly. See the Key Scheme Table in Appendix A of the *payShield 9000 Host Programmer's Manual*.

List of Host Commands (Alphabetical)

List of Host Commands (Functional)

Chapter 2 – Host Commands

General

This Chapter details all the commands available with their responses and possible error codes.

A number of abbreviations are used throughout. They are:

L	:	Encrypted PIN length. Set at installation.
m	:	Message Header length. Set at installation.
n	:	Variable length field.
A	:	Alphanumeric (can include any non-control type) characters.
H	:	Hexadecimal character ('0'...'9', 'A'...'F').
N	:	Numeric Field ('0'...'9').
C	:	Control character.
B	:	Binary data (byte) (X'00...X'FF).
D	:	Binary coded decimal (BCD) character ('0'...'9').

For example:

32 H : Indicates that thirty-two hexadecimal characters are required.

m A : Indicates the string of "message header length" alphanumeric characters.

For convenience, the STX and ETX control characters, which bracket every command and response when using Asynchronous communications, are not shown in the details that follow.

In a command to the HSM, any key can be replaced by a reference to internal user storage. In the details that follow, a key is always shown as if it is to be sent with each command; in every case the key can be replaced by the index flag K and a three-digit pointer value.

The HSM can be used in systems where there may be Atalla security equipment at other network nodes. This is achieved by the inclusion of an Atalla variant in those commands that translate a key from/to encryption under a ZMK. This has the effect of modifying the ZMK before it is used to decrypt/encrypt in accordance with the method used by the Atalla equipment. The HSM can support 1 or 2 digit Atalla variants.

When a disabled host command is invoked, the error code 68 is returned.

Common Key Management Commands

Use of these key management commands requires the optional Contactless or EMV-based Cards Data Preparation license.

The HSM provides the following host commands to support key management operations for Contactless and EMV-based Data Preparation:

Derive Card Unique DES Keys

Variant LMK ☑	Key Block LMK ☑
License: **HSM9-LIC011** or **HSM9-LIC016**	
Authorization: **Not required**	

Function: Derive a Card Unique Key from a Master Key.

Notes: This function supports:

- Generation of Unique Derived keys for EMV cards
- Generation of Card Master Keys for personalization using EMV CPS
- Generation of the Account Unique Keys AUK DCVV for Discover

This function supports generation of session keys for cloud based mobile payment schemes or HCE:

- Visa Limited Use keys
- AMEX Limited Use keys
- Discover Financial Services One Time Personalisation keys

As shown in the table below, the Key Derivation Method determines the algorithm for the Master Key, Card Key & KEK, and additionally the scheme that can be used for encrypting the derived key under the KEK.

Key Derivation Method	Master Key, Card Key & KEK Algorithm	Key Scheme (KEK)
'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', '2', '3'	3DES	'X' for a 112-bit key 'Y' for a 168-bit key (both ANSI X9.17)
'I'	AES	'N' for any AES key (NIST SP800-38F Key Wrap)

Key Derivation Method	MK Key Type (Variant LMK)	MK Key Usage (Key Block LMK)
'A', 'B', 'F'	'109', '209', '309', '509', '709', '207'	'E0', 'E2', 'E1', 'E4', 'E6', '32', 'E7'
'2', '3'	'709'	'32'
'C', 'D'	'207'	'E7'
'E', 'H'	'109'	'E0'
'G' (OTPK Key Gen = 0 or 1)	'109'	'E0'
'G' (OTPK Key Gen = 2 or 3)	'709'	'E6', 'E2'
'G' (OTPK Key Gen = 4 or 5)	'402'	'C0', '12', '13'
'I'	n/a	'E0', 'E2', 'E1', 'E4', 'E6', '32', 'E7'

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">COMMAND MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Subsequently returned to the Host unchanged.</td></tr>
<tr><td>Command Code</td><td>2 A</td><td>Value 'KI'.</td></tr>
<tr><td>MK Type</td><td>3 H</td><td>For a Variant LMK, the field specifies the Master Key Type. The following Key Types are permitted:
'109': MK-AC encrypted under LMK 28-29/1
'209': MK-SMI encrypted under LMK 28-29/2
'309': MK-SMC encrypted under LMK 28-29/3
'509': MK-DN encrypted under LMK 28-29/5
'709': MK-CVC3 or MK-DCVV encrypted under LMK 28-29/7
'207': KMC encrypted under LMK 24-25/2
'402': CVK encrypted under LMK 14-15/4.
For a Key Block LMK, this field is ignored and should be set to 'FFF'.</td></tr>
<tr><td>MK</td><td></td><td>Master Key from which the unique key is derived, encrypted under LMK.</td></tr>
<tr><td></td><td>32 H
or
'U' + 32 H
or
'T' + 48 H</td><td>For a Variant LMK, the 'MK' must be encrypted under the appropriate LMK pair/variant defined by the 'MK Type' field.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a Key Block LMK, the 'MK' must comply with one of the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'12', '13', 'C0', 'E1', 'E4', 'E7'</td><td>'T'</td><td>'N'</td></tr>
<tr><td>'32', 'E0', 'E2', 'E6'</td><td>'T', 'A'</td><td>'N'</td></tr>
</table></td></tr>
<tr><td>Key Derivation Method</td><td>1 A</td><td>EMV methods to be used to derive a unique key from the Master Key (See Reference 6):
'A': EMV 4.1 Book 2 Option A method
'B': EMV 4.1 Book 2 Option B method
'C': EMV CPS method for Card Master Keys
Note: This derivation method will derive 3 keys (CK-ENC, CK-MAC and CK-DEK) from the Master key as defined in EMV CPS.
'D': Generic Personalisation Key Derivation Method
'E': Visa Limited Use Key (LUK) (only valid if MK Type = 109)
'F': AMEX Limited Use Key (LUK)
'G': Discover One Time Personalisation Key (OTPK)
'H': Visa Limited Use Key (LUK) QR Code (only valid if MK Type = 109)
'I': EMV 4.3 Option 'C' method (AES)
Discover Network specific methods for deriving unique key from Master Key (only valid if MK Type = 709):
'2': 16-byte Account Unique Key for DCVV (AUK_{DCVV})
'3': 24-byte Account Unique Key for DCVV (AUK_{DCVV})</td></tr>
<tr><td>Derivation Data</td><td>n N</td><td>If Key Derivation Method = 'A', 'B', 'E', 'F', 'H' or 'I':
Concatenation of the Primary Account Number and 2 digit Sequence Number for the card.
If the Sequence Number is not available it should be specified as '00'.</td></tr>
<tr><td></td><td>6 B</td><td>If Key Derivation Method = 'C':
Card Key derivation data (e.g. KEYDATA composed of Master Key ID and Chip Serial Number)</td></tr>
<tr><td></td><td>16 B</td><td>If Key Derivation Method = 'D':
The key derivation data.</td></tr>
<tr><td></td><td>20 N</td><td>If Key Derivation Method = '2' or '3', or Key Derivation</td></tr>
</table>

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td></td><td></td><td>Method = 'G' and OTPK Key Generation Method = '2' or '3':
Concatenation of the 16 digit Primary Account Number (PAN) and 4 digit Expiration Date for the card.</td></tr>
<tr><td></td><td>n N</td><td>If Key Derivation Method = 'G' and OTPK Key Generation Method = '0', '1', '4' or '5':
Concatenation of the Primary Account Number and 2 digit PAN Sequence Number. If the Sequence Number is not available it should be specified as '00'.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Delimiter. Value ';'.</td></tr>
<tr><td colspan="3">If Key Derivation Method = 'E' or 'H', the following field will be present:</td></tr>
<tr><td>YHHHHCC</td><td>7 N</td><td>Only present if Key Derivation Method = 'E'.
The Year/Hour/Counter value used to derive the Limited Use Key (LUK) to produce the Limited Use Cryptogram (LUC).
Y (0-9) : least significant digit of current year
HHHH (0001-8784) : number of hours since Jan 1st
CC (00-99) : counter</td></tr>
<tr><td colspan="3">If Key Derivation Method = 'F', the following 4 fields must be present:</td></tr>
<tr><td>Output Key Flag</td><td>1 A</td><td>'0': Derive SK-AC session key
'1': Derive SK-CVM session key</td></tr>
<tr><td>UDK Key Derivation Method</td><td>1 A</td><td>'A': EMV 4.3 ii Option A 3DES ECB
'B': EMV 4.3 ii Option B 3DES ECB</td></tr>
<tr><td>Session Key Method</td><td>2 N</td><td>'01': EMV Common Session Key Derivation</td></tr>
<tr><td>ATC</td><td>2 B</td><td>Application Transaction Counter</td></tr>
<tr><td colspan="3">If Key Derivation Method = 'F' and Output Key Flag = '1', the following 3 fields must be present:</td></tr>
<tr><td>Passcode Key Method</td><td>2 N</td><td>'01': PBKDF2 with HMAC SHA-1</td></tr>
<tr><td>CVM Session Key Method</td><td>2 N</td><td>'01': 3DES CBC</td></tr>
<tr><td>IV</td><td>8 B or 16 B</td><td>Initialisation Vector.
8 B: If CVM Session Key Method = '01'
16 B: If CVM Session Key Method = '02'</td></tr>
<tr><td colspan="3">If Key Derivation Method = 'F', Output Key Flag = '1' and Passcode Key Method = '01', the following 7 fields must be present:</td></tr>
<tr><td>PIN Block Format Code</td><td>2 N</td><td>Valid PIN block format codes are:
'02': Docutel ATM format
'03': Diebold & IBM ATM format
'05': ISO 9564-1 format 1
'34': Standard EMV 1996 format</td></tr>
<tr><td>Passcode PIN Block</td><td>8 B</td><td>The Passcode PIN Block 3DES ECB encrypted under the ZPK.</td></tr>
<tr><td>ZPK</td><td></td><td>The Zone PIN Key, encrypted under the LMK.</td></tr>
<tr><td></td><td>32 H
or
'U' + 32 H
or
'T' + 48 H</td><td>For a Variant LMK, the ZPK encrypted under LMK pair 06-07.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a LMK key block, the ZPK encrypted under LMK should comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode Of Use</th></tr>
<tr><td>'P0', '72'</td><td>'T', 'A'</td><td>'B', 'D', 'N'</td></tr>
</table></td></tr>
<tr><td>Salt Flag</td><td>1 N</td><td>'0': Random salt supplied in command
'1': Random salt generated by command</td></tr>
</table>

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td>Salt Length</td><td>2 N</td><td>The length of the Salt supplied or the length of the Salt to generate.</td></tr>
<tr><td>Salt</td><td>n B</td><td>If Salt Flag is '0', the random salt value.</td></tr>
<tr><td>Iteration Count</td><td>6 N</td><td>The number of iterations performed by the PBKDF. Minimum value 1000.</td></tr>
<tr><td colspan="3">If Key Derivation Method = 'G', the following fields must be present:</td></tr>
<tr><td>Number of OTPKs</td><td>2 N</td><td>The number of OTPKs to generate.</td></tr>
<tr><td>OTPK Key Generation Method</td><td>1 A</td><td>The OTPK Key Generation Method:
'0': Default EMV using MK AC
'1': Server PIN based white box using MK AC
'2': Enhanced DCVV OTPK using AUK DCVV
'3': Server PIN based white box using AUK DCVV
'4': Default EMV using CVK
'5': Server PIN based white box using CVK</td></tr>
<tr><td colspan="3">If OTPK Key Generation Method = '1', '3' or '5', the following 3 fields must be present:</td></tr>
<tr><td>PIN Block Format Code</td><td>2 N</td><td>Valid PIN block format codes are:
'02': Docutel ATM format
'03': Diebold & IBM ATM format
'05': ISO 9564-1 format 1
'34': Standard EMV 1996 format</td></tr>
<tr><td>PIN Block</td><td>8 B</td><td>The PIN Block encrypted under the ZPK.</td></tr>
<tr><td>ZPK</td><td></td><td>The Zone PIN Key, encrypted under the LMK.</td></tr>
<tr><td></td><td>32 H
or
'U' + 32 H
or
'T' + 48 H</td><td>For variant LMK, the ZPK is encrypted under LMK pair 06-07.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For Key Block LMK, the ZPK must comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode Of Use</th></tr>
<tr><td>'P0', '72'</td><td>'T', 'A'</td><td>'B', 'D', 'N'</td></tr>
</table></td></tr>
<tr><td colspan="3">For all OTPK Key Generation Methods, the following 2 fields must be present:</td></tr>
<tr><td>Start ATC</td><td>2 B</td><td>Application Transaction Counter, n.</td></tr>
<tr><td>Key Transport Algorithm</td><td>2 N</td><td>'00': OTPKs encrypted using AES ECB mode</td></tr>
<tr><td colspan="3">For all Key Derivation Methods, the following fields must be present:</td></tr>
<tr><td>KEK</td><td></td><td>The transport Key Encryption Key, encrypted under the LMK.</td></tr>
<tr><td></td><td>32 H
or
'U' + 32 H
or
'T' + 48 H</td><td>For a Variant LMK, the 'KEK' is encrypted under LMK pair 24-25 variant 1.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a Key Block LMK, the 'KEK' must comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode Of Use</th></tr>
<tr><td>'54'</td><td>'T', 'A'</td><td>'B', 'D', 'E', 'N'</td></tr>
</table></td></tr>
<tr><td>Key Scheme (KEK)</td><td>1 A</td><td>Key scheme for encrypting derived key under KEK.
For a 3DES KEK, this field must be 'X' or 'Y' (ANSI X9.17).
For an AES KEK, this field must be 'N' (NIST SP800-38F Key Wrap).</td></tr>
<tr><td>Key Check Value Type</td><td>1 A</td><td>'0': KCV for entire key, 6 bytes long
'1': KCV for entire key, 3 bytes long</td></tr>
<tr><td>Atalla Variant</td><td>1 / 2 N</td><td>Optional. Atalla variant; for use in systems with Atalla equipment.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value '%'. Optional; if present, the following field must be present.</td></tr>
</table>

Field	Length & Type	Details
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by license; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'KJ'.
Error Code	2 N	'00': No error '04': Unrecognized key derivation method '05': Invalid MK type '06': Invalid MK type or Derivation Data for Derivation Method or YHHHHCC value '10': MK parity error '11': KEK parity error '20': Invalid PIN Block '23': Invalid PIN Block Format value '27': Key Scheme (KEK) and derived key mismatch 'DA': KEK – key block error 'DB': ZPK – key block error 'E2': ZPK parity error 'EA': MK – key block error 'F1': Invalid UDK Method 'F2': Invalid Session Key Method 'F3': Invalid CVM Session key Method 'F4': Invalid Passcode Key Method 'F5': Invalid Salt Flag value 'F6': Invalid Output Key Flag 'F7': Invalid Iteration Count 'F8': Invalid OTPK Key Generation Method value 'F9': Invalid Key transport Algorithm 'FA': Invalid KEK Key Type or Algorithm 'FB': Invalid OTPK Count 'FC': Invalid Derivation Data length 'FD': Invalid OTPK Option 'FE': Not supported with variant LMK 'FF': Key Derivation Method not allowed or a standard error code.
If Error Code = 'DA', 'DB' or 'EA', the following field will be present:		
Additional Error Code	2 A	The key block specific error code.
If Key Derivation Method = 'A', 'B', 'E', 'I', '2' or '3', the following 2 fields will be present:		
DK (KEK)	'X' + 16 B or 'Y' + 24 B	The derived unique key encrypted under KEK.
Key check value	6 B or 3 B	The key check value for the DK.
If Key Derivation Method = 'C', the following 6 fields will be present:		
CK-ENC	'X' + 16 B	Card Key for cryptograms encrypted under KEK.
CK-ENC KCV	6 B or 3 B	The key check value for K-ENC.
CK-MAC	'X' + 16 B	Card Key for authentication encrypted under KEK.
CK-MAC KCV	6 B or 3 B	The key check value for K-MAC.
CK-DEK	'X' + 16 B	Card Key for encryption encrypted under KEK.
CK-DEK KCV	6 B or 3 B	The key check value for K-DEK.
If Key Derivation Method = 'D', the following 2 fields will be present:		
PSK	'X' + 16 B	The Personalisation System Key encrypted under the KEK
PSK KCV	6 B or 3 B	The PSK key check value
If Key Derivation Method = 'F' and Output Key Flag = '0', the following 2 fields will be present:		
SK-AC	'X' + 16 B	The AC session key encrypted under the KEK.
SK-AC KCV	6 B or 3 B	The AC session key check value.

Field	Length & Type	Details
If Key Derivation Method = 'F' and Output Key Flag = '1', the following 3 fields will be present:		
SK-CVM	'X' + 16 B	The Passcode protected CVM session key encrypted under the KEK.
SK-CVM KCV	6 B or 3 B	The CVM session key check value.
Salt Value	n B	If Salt Flag is '1', the generated random salt value.
If Key Derivation Method = 'G', the following fields will be present:		
Final ATC	2 B	The updated ATC value, n + Number Of OTPKs.
Key Blob Length	4 N	The length in bytes of the following key blob field.
For the number of OTPKs requested, the key blob will consist of the concatenation of:		
SK	'X' + 16 B	The OTPK session key encrypted under the KEK
SK KCV	6 B or 3 B	The OTPK session key encrypted under the KEK
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Export a Key under a KEK

Variant LMK ☑	Key Block LMK ☑
License: **HSM9-LIC011** or **HSM9-LIC016**	
Authorization: **Not required**	

Function: Encrypt a Key under a KEK for transport to EMV data personalization system

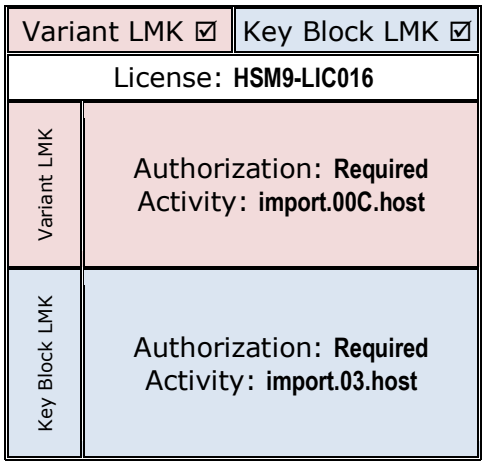
Notes: See the Key Scheme Table in Appendix A of the *payShield 9000 Host Programmer's Manual* for schemes available to encrypt keys.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'K8'.
Key Type	3 H	For a Variant LMK, this is the type of the key to be exported. Only the following Key Types are permitted: '001': ZPK encrypted under LMK 06-07 '008': ZAK encrypted under LMK 26-27 '00A': ZEK encrypted under LMK 30-31 *This field can be expanded in the future to accommodate other key types.*
		For a Key Block LMK, this field is ignored and should be set to 'FFF'.
Key		The key to be exported, encrypted under the LMK.
	16 H or 'U' + 32 H or 'T' + 48 H	For a Variant LMK, the Key is encrypted under the LMK pair/variant defined by Key Type.
	or 'S' + n A	For a Key Block LMK, the Key in key block format; must comply with the following: **Key Usage**: 'P0', '72' / 'M1', 'M2', 'M3' / 'D0', '22' **Algorithm**: 'T' **Mode Of Use**: 'B', 'D', 'E', 'N' / 'C', 'G', 'N', 'V' / 'B', 'D', 'E', 'N'
KEK		The Key Encryption Key, to be used to protect the exported key.
	32 H or 'U' + 32 H or 'T' + 48 H	For a Variant LMK, the Key is encrypted under LMK 24-25/1.
	or 'S' + n A	For a Key Block LMK, the Key in key block format; must comply with the following: **Key Usage**: '54' **Algorithm**: 'T' **Mode Of Use**: 'N'
Key Scheme (KEK)	1 A	Key scheme for encrypting key under KEK. Valid values are 'X', 'Y', 'R', 'Z', 'U' and 'T'. See the Key Scheme Table in Appendix A of the *payShield 9000 Host Programmer's Manual*.
Atalla Variant	1 / 2 N	Optional. Atalla variant; for use in systems with Atalla equipment.
Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.

Field	Length & Type	Details
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
If Variant LMK and Key Scheme (KEK) = 'R', the following fields must be present:		
Delimiter	1 A	Mandatory. Must be '&'
Key Usage	2 A	Key Usage field, to be included in the TR-31 key block header; any permitted value for the key type. For a list of possible values, see the Key Usage Table in Chapter 8 of the *payShield 9000 Host Programmer's Manual*.
Mode Of Use	1 A	Mode of Use field, to be included in the TR-31 key block header; any permitted value for the key type. For a list of possible values, see the Mode of Use Table in Chapter 8 of the *payShield 9000 Host Programmer's Manual*.
Key Version Number	2 N	Key Version Number field, to be included in the TR-31 key block header; permitted values: '00' to '99'.
Exportability	1 A	Exportability field, to be included in the TR-31 key block header. The only permitted values are 'N' or 'S';
Number of Optional Blocks	2 N	Number of Optional Blocks specified below, to be included in the TR-31 key block header; permitted values '00' to '02'.
If Number of Optional Blocks > '00', the following 3 fields are repeated for each optional block.		
Optional Block Identifier	2 A	Optional Block Identifier; to be included in the TR-31 key block header; any permitted value except 'PB'.
Optional Block Length	2 H	Number of characters in the optional block (including identifier and length); to be included in the TR-31 key block header; permitted values X'04 to X'FF; if value = X'04, then the following field is not present.
Optional Block Data	n A	Optional block data; to be included in the TR-31 key block header.
Delimiter	1 A	Value '!'. Optional. If present, the following field must be present.
Key Block Version ID	1 A	Key Block Version ID: 'A': Key block protected using the Key Variant Binding Method (see Ref 2 section 5.3.3.1) 'B': Key block protected using the Key Derivation Binding Method (see Ref 2 section 5.3.2.1) 'C': Key block protected using the Key Variant Binding Method (see Ref 2 section 5.3.3.1) 'D' – Key block protected using the AES Key Derivation Binding Method (see Ref 8 section 5.3.2.3).
If Key Block LMK and Key Scheme = 'R', then the following fields must be present:		
Delimiter	1 A	Mandatory. Must be '&'.
Modified Exportability	1 A	Exportability field, to be included in the key block header. The only permitted value = 'N'.
Delimiter	1 A	Value '!'. Optional; can only be present when exporting the key to TR-31 keyblock format. If present, the following field must be present.
Key Block Version ID	1 A	'A': Key block protected using the Key Variant Binding Method (see Ref 2 section 5.3.3.1) 'B': Key block protected using the Key Derivation Binding Method (see Ref 2 section 5.3.2.1) 'C': Key block protected using the Key Variant Binding Method (see Ref 2 section 5.3.3.1) 'D' – Key block protected using the AES Key Derivation Binding Method (see Ref 8 section 5.3.2.3).
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'K9'.
Error Code	2 N	'00': No error '05': Invalid key type '10': Key parity error '11': KEK parity error 'DA': KEK – key block error 'DB': Key – key block error or a standard error code.
If Error Code = 'DA' or 'DB', the following field will be present		
Additional Error Code	2 A	The key block specific error code.
Key (KEK)	8 B or 'X' + 16 B or 'U' + 16 B or 'Y' + 24 B or 'T' + 24 B or 'R' + n A	The exported key, encrypted under the KEK.
Key check value	3 B	The key check value.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Import an RSA Private Key



Function: This function supports the import an RSA Private Key from encryption under a Zone Master Key to encryption under the LMK.

The comparable algorithm key strengths defined in NIST SP800-57 part 1 are applied when importing or exporting a key under a ZMK.

The following security settings must be configured to allow use of this command:

- **Enable import and export of RSA Private Keys:** Yes
- **Key export and import in trusted format only:** No

Notes: See: *payShield 9000 Host Programmer's Manual: The RSA Cryptosystem* for more information.

The RSA private key for import must be in CRT format (p, q, dp, dq, u) where dp= d mod (p-1), dq= d mod (q-1), u = (q-1) mod p.

Each CRT component is decrypted using an AES or 3DES ZMK, and block cipher mode ECB or CBC.

The following table indicates the size of 3DES or AES key required to protect the different size RSA private keys:

Key Length (bits)	3DES ZMK	AES ZMK
320..1024	112/168-bit	128/192/256-bit
1025...2048	168-bit only	128/192/256-bit
2049...3072	Not allowed	128/192/256-bit
3073...4096	Not allowed	192/256-bit only

The following table indicates the possible values for the output key block, containing the RSA private key:

Key Usage	Valid Mode of Use values
'03' (for signing/key mgt)	'D', 'N', 'S'
'04' (for ICCs)	'N', 'S'
'05' (for PIN translation)	'D', 'N'
'06' (for data decryption)	'D', 'N'

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">COMMAND MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Subsequently returned to the Host unchanged.</td></tr>
<tr><td>Command Code</td><td>2 A</td><td>Value 'L6'.</td></tr>
<tr><td>ZMK</td><td></td><td>The Zone Master Key, used to protect the exported RSA private key.</td></tr>
<tr><td></td><td>32 H
or
'U' + 32 H
or
'T' + 48 H</td><td>For a Variant LMK, the ZMK is encrypted under LMK 04-05 variant 0.</td></tr>
<tr><td></td><td>or
'S' + n B</td><td>For a Key Block LMK, the ZMK must comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'K0', 'K1', '52'</td><td>'T', 'A'</td><td>'B', 'D', 'E', 'N'</td></tr>
</table></td></tr>
<tr><td>Key Format</td><td>1 N</td><td>The format of the private key to import.
'0': CRT components (P, Q, DP, DQ, U in order)</td></tr>
<tr><td>Block Cipher Mode</td><td>1 N</td><td>The Block Cipher Mode to use when decrypting the RSA private key under the ZMK:
'0': ECB
'1': CBC. Each component (plus any pad) individually encrypted using an IV of 0.</td></tr>
<tr><td>Pad Mode</td><td>1 N</td><td>The ISO 9797-1 padding mode:
'1': Optional 0x00 pad bytes to block length appended
'2': Mandatory 0x80 and optional 0x00 to block length appended
'3': Prepended length byte(s) (containing the unpadded CRT component byte length value) and optional 0x00 pad appended</td></tr>
<tr><td colspan="3">If Pad Mode = '3', the following field must be present:</td></tr>
<tr><td>Length Byte</td><td>1 N</td><td>'0': non BER encoded length (one byte)
'1': BER encoded length.</td></tr>
<tr><td>Private Key Length</td><td>4 H</td><td>The length of the Private Key in bytes</td></tr>
<tr><td>Private Key</td><td>n B</td><td>Private Key encrypted under the ZMK</td></tr>
<tr><td>Validate imported private key flag</td><td>1 N</td><td>'0': No validation of the private key is performed.
'1': Validate the imported private key</td></tr>
<tr><td colspan="3">If Validate Imported Private key flag = '1', the following 4 fields must be present</td></tr>
<tr><td>Public Key length</td><td>4 H</td><td>The length of the Public Key modulus in bytes</td></tr>
<tr><td>Public key</td><td>n B</td><td>The public key modulus (n)</td></tr>
<tr><td>Public Exponent Length</td><td>4 H</td><td>The length of the Public Exponent in bytes</td></tr>
<tr><td>Public Exponent</td><td>n B</td><td>The exponent (e)</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value '%'. Optional; if present, the following field must be present.</td></tr>
<tr><td>LMK Identifier</td><td>2 N</td><td>LMK identifier; min value = '00'; max value is defined by licence; must be present if the above Delimiter is present.</td></tr>
</table>

Field	Length & Type	Details
If Key Block LMK, the following fields must be present:		
Delimiter	1 A	Value '#'.
Delimiter	1 A	Value '~'. Optional; if present, the following 2 fields must be present.
Key Usage	2 H	Only present if the '~' delimiter (above) is present. Key Usage field, to be included in the key block header; permitted values: '03', '04', '05', '06'. If not present, the imported key will have a Key Usage field of '03'.
Mode of Use	1 A	Only present if the '~' delimiter (above) is present. Mode of Use field, to be included in the key block header; permitted values 'D', 'N', 'S'. If not present, the imported key will have a Mode of Use field of 'N'.
Key Version Number	2 N	Key Version Number field, to be included in the key block header; permitted values: '00' to '99';
Exportability	1 A	Exportability field, to be included in the key block header; only permitted values are 'N', 'E' or 'S';
Number of Optional Blocks	2 N	Number of Optional Blocks specified below; permitted values '00' to '08';
For each optional block, the following three fields must be specified. Note: if the Number of Optional Blocks = '00', then none of the following three fields should be present.		
Optional Block Identifier	2 A	Optional Block Identifier; any permitted value except 'PB'.
Optional Block Length	2 H	Number of characters in the optional block (including identifier and length); permitted values X'04 to X'FF; if value = X'04, then the following field is not present
Optional Block Data	n A	Optional block data
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">RESPONSE MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Returned to the Host unchanged.</td></tr>
<tr><td>Response Code</td><td>2 A</td><td>Value 'L7'.</td></tr>
<tr><td>Error Code</td><td>2 A</td><td>'00': No error
'03': Command disabled by security configuration
'10': ZMK key parity error
'80': Private Key Length error
'D1': ZMK – key block error
'D3': Invalid Key Format
'D4': Invalid format for private key
'D5': ZMK key strength error (key length security check failed)
'D6': Key export not allowed (Export field in key block set to 'N')
'D7': Private Key – key block error
'D8': Invalid Block Cipher Mode
'D9': Invalid pad mode
'DA': Invalid length byte value
'DB': Public key length error
'DC': Public Exponent length error
'DD': Imported CRT length error
'DE': Invalid Validate Imported Secret Key Flag
or a standard error code.</td></tr>
<tr><td colspan="3">If Error Code = 'D1' or 'D7', the following field will be present:</td></tr>
<tr><td>Additional Error Code</td><td>2 A</td><td>The key block specific error code.</td></tr>
<tr><td>Private Key Length</td><td>4 H</td><td>Only present if using a Variant LMK.
If present, this field indicates the length of the following field.</td></tr>
<tr><td rowspan="3">Private Key</td><td></td><td>The RSA private key, encrypted under the LMK.</td></tr>
<tr><td>n B</td><td>For a Variant LMK, the Private Key is encrypted under LMK 34-35 variant 0.</td></tr>
<tr><td>or
'S' + n B</td><td>For a Key Block LMK, the Private Key will conform to:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'03' or Key Usage specified after '~' delimiter</td><td>'R'</td><td>'S' or Mode of Use specified after '~' delimiter</td></tr>
</table></td></tr>
<tr><td>End Message Delimiter</td><td>1 C</td><td>Present only if present in the command message. Value X'19.</td></tr>
<tr><td>Message Trailer</td><td>n A</td><td>Present only if present in the command message. Maximum length 32 characters.</td></tr>
</table>

Export an RSA Private Key

Variant LMK ☑	Key Block LMK ☑
License: **HSM9-LIC016**	
Variant LMK	Authorization: **Required** Activity: **export.00C.host**
Key Block LMK	Authorization: **Required** Activity: **export.03.host**

Function: This function supports the export of an RSA Private Key from encryption under the LMK to encryption under a Zone Master Key.

The comparable algorithm key strengths defined in NIST SP800-57 part 1 are applied when importing or exporting a key under a ZMK.

The following security settings must be configured to allow use of this command:

- 'Enable import / export of RSA Private Keys?' MUST be set to 'YES' (defaults to NO)
- 'Key export and import in trusted format only?' MUST be set to 'NO' (defaults to YES)

Notes: See: *payShield 9000 Host Programmer's Manual: The RSA Cryptosystem* for more information.

The exported RSA private key will be in CRT format (p, q, dp, dq, u) where dp= d mod (p-1), dq= d mod (q-1), u = (q-1) mod p.

Each CRT component is encrypted using an AES or 3DES ZMK, and block cipher mode ECB or CBC.

The following table indicates the size of 3DES or AES key required to protect the different size RSA private keys:

Key Length (bits)	3DES ZMK	AES ZMK
320..1024	112/168-bit	128/192/256-bit
1025...2048	168-bit only	128/192/256-bit
2049...3072	Not allowed	128/192/256-bit
3073...4096	Not allowed	192/256-bit only

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">COMMAND MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Subsequently returned to the Host unchanged.</td></tr>
<tr><td>Command Code</td><td>2 A</td><td>Value 'L8'.</td></tr>
<tr><td>ZMK</td><td></td><td>The Zone Master Key, used to protect the exported RSA private key.</td></tr>
<tr><td></td><td>32 H
or
'U' + 32 H
or
'T' + 48 H</td><td>For a Variant LMK, the ZMK is encrypted under LMK 04-05.</td></tr>
<tr><td></td><td>or
'S' + n B</td><td>For a Key Block LMK, the ZMK must comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'K0', 'K1', '52'</td><td>'T', 'A'</td><td>'B', 'D', 'E', 'N'</td></tr>
</table></td></tr>
<tr><td>Key Format</td><td>1 N</td><td>The format of the private key to export.
'0': CRT components (P, Q, DP, DQ, U in order)</td></tr>
<tr><td>Block Cipher Mode</td><td>1 N</td><td>The Block Cipher Mode to be used to encrypt the RSA private key under the ZMK:
'0': ECB
'1': CBC. Each component (plus any pad) individually encrypted using an IV of 0.</td></tr>
<tr><td>Pad Mode</td><td>1 N</td><td>The ISO 9797-1 padding mode:
'1': Optional 0x00 pad bytes to block length appended
'2': Mandatory 0x80 and optional 0x00 to block length appended
'3': Prepended length byte(s) (containing the unpadded CRT component byte length value) and optional 0x00 pad appended</td></tr>
<tr><td colspan="3">If Pad Mode = '3', the following field must be present:</td></tr>
<tr><td>Length Byte</td><td>1 N</td><td>'0': non BER encoded length (one byte)
'1': BER encoded length.</td></tr>
<tr><td>Private Key Length</td><td>4 H</td><td>The length of the Private Key in bytes
For a Variant LMK, the length of the Private Key, in bytes.
For a Key Block LMK, this field is ignored and should be set to 'FFFF'.</td></tr>
<tr><td>Private Key</td><td></td><td>The Private Key, encrypted under the LMK.</td></tr>
<tr><td></td><td>n B</td><td>For a Variant LMK, the Private Key is encrypted under LMK pair 34-35 variant 0.</td></tr>
<tr><td></td><td>or
'S' + n B</td><td>For a Key Block LMK, the Private Key must comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'03'</td><td>'R'</td><td>'S', 'D', 'N'</td></tr>
</table></td></tr>
<tr><td>End Message Delimiter</td><td>1 C</td><td>Must be present if a message trailer is present. Value X'19.</td></tr>
<tr><td>Message Trailer</td><td>n A</td><td>Optional. Maximum length 32 characters.</td></tr>
</table>

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'L9'.
Error Code	2 A	'00': No error '03': Command disabled by security configuration '10': ZMK key parity error '75': Public/Private key mismatch '80': Private Key Length error 'D1': ZMK – key block error 'D3': Invalid Key Format 'D4': Invalid format for private key 'D5': ZMK key strength error (key length security check failed) 'D6': Key export not allowed (Export field in key block set to 'N') 'D7': Private Key – key block error 'D8': Invalid Block Cipher Mode 'D9': Invalid pad mode 'DA': Invalid length byte value 'DB': Public key length error 'DC': Public Exponent length error 'DD': Imported CRT length error 'DE': Invalid Validate Imported Secret Key Flag or a standard error code.
If Error Code = 'D1' or 'D7', the following field will be present:		
Additional Error Code	2 A	The key block specific error code.
If Error Code = '00', the following 2 fields will be present:		
Private Key Length	4 H	The length of the Private Key.
Private Key	n B	The exported Private Key, encrypted under the ZMK.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Generate Digitized Card Single Use Keys

Variant LMK ☒	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Not required**	

Function: Derive Single Use Keys (SUKs) for Host Card Emulation (HCE) mobile payment applications.

Notes:

Generate Digitized Card Single Use Keys (Mode Flag '0': MCBP SUK Derivation; include PIN; output in legacy JSON format.)

Variant LMK ☒	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Not required**	

Function: Derive Single Use Keys (SUKs) for Mastercard Cloud Based Payments (MCBP). The user and mobile PINs will be used during the production of the SUKs. The keys will be output in legacy JSON format.

Notes: This command generates one or more DC_SUK blocks that contain the single use keys for user and mobile device authentication for contactless and remote payments, the ATC and the Dynamic Number for use in CDA.

For details of the JSON format used in the DC_SUK block, refer to Appendix O – DC_SUK block template.

The DC_SUK blocks are concatenated together and encrypted using the Remote Management Session key for confidentiality MS_KEY_CONF and MAC protected using the Remote Management Session key for integrity MS_KEY_MAC. The output is in the format suitable to loading directly to the Mobile Payment Application.

The ATC value is updated by this command according to the number of DC_SUK blocks generated.

The following input data elements will have been generated by the data preparation system using the 'KI' Generate Card Unique Keys function encrypted under a transport key:

- CMK_CL_UMD
- CMK_RP_UMD
- CMK_CL_MD
- CMK_RP_MD
- CMK_IDN

The transport keys will have been imported and encrypted under the Local Master Key.

The Mobile PIN block will be created using the PIN Translation commands and transported encrypted under a ZPK.

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">COMMAND MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Subsequently returned to the Host unchanged.</td></tr>
<tr><td>Command Code</td><td>2 A</td><td>Value 'IY'.</td></tr>
<tr><td>Mode Flag</td><td>1 N</td><td>'0': MCBP SUK Derivation; include PIN; output in legacy JSON format</td></tr>
<tr><td>Key Generation Option Delimiter</td><td>1 A</td><td>Value '$'. Optional; if present, the following field must be present.</td></tr>
<tr><td>Key Generation Option</td><td>1 N</td><td>'0': Generate SUKs for Contactless Transactions (with PIN)
'1': Generate SUKs for Remote Payment Transactions (with PIN)
'2': Generate SUKs for Contactless Transactions (no PIN)
'3': Generate SUKs for Remote Payment Transactions (no PIN)
'4': Generate SUKs for Contactless & Remote Payment Transactions (no PIN)</td></tr>
<tr><td>Message Counter</td><td>5 N</td><td>The message counter is maintained by the CMS and is incremented for each message sent from the CMS to MPA.
Maximum value is 65535 or 0x00FFFF.</td></tr>
<tr><td>MS_KEY_CONF</td><td>'S' + n A</td><td>The Remote Management Session Key for message confidentiality, encrypted under AES Key Block LMK and which must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'35'</td><td>'A'</td><td>'B', 'D', 'E', 'N'</td></tr>
</table></td></tr>
<tr><td>MS_KEY_MAC</td><td>'S' + n A</td><td>The Remote Management Session Key for message integrity, encrypted under AES Key Block LMK and which must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'36'</td><td>'A'</td><td>'C', 'G', 'V', 'N'</td></tr>
</table></td></tr>
<tr><td>Start ATC</td><td>2 B</td><td>The start value for the Application Transaction Counter</td></tr>
<tr><td>Number of DC_SUK</td><td>2 N</td><td>The number of DC_SUK blocks to generate.
Value: '01'…'50'.</td></tr>
<tr><td>CMK_CL_UMD</td><td>'X' + 16 B</td><td>This field is NOT present when Key Generation Option = '1' or '3'.
User and Mobile Device Authentication Key for contactless transactions. 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>CMK_RP_UMD</td><td>'X' + 16 B</td><td>This field is NOT present when Key Generation Option = '0' or '2'.
User and Mobile Device Authentication Key for remote payments. 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>CMK_CL_MD</td><td>'X' + 16 B</td><td>This field is NOT present when Key Generation Option = '1' or '3'.
Mobile Device Authentication Key for contactless transactions. 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>CMK_RP_MD</td><td>'X' + 16 B</td><td>This field is NOT present when Key Generation Option = '0' or '2'.
User and Mobile Device Authentication Key for remote payments. 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>CMK_IDN</td><td>'X' + 16 B</td><td>CDA Dynamic Number generation key encrypted 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>Atalla Variant</td><td>2 N</td><td>Atalla variant; for use in systems with Atalla equipment.
Must be set to '00' if not used.</td></tr>
<tr><td>Transport KEK</td><td>'S' + n A</td><td>The transport KEK key encrypted under LMK key block and should comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'54'</td><td>'T', 'A'</td><td>'B', 'D', 'E', 'N'</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">If Key Generation Option = '2', '3' or '4', the following 6 fields are omitted:</td></tr>
<tr><td>Mobile PIN Block Length</td><td>2 N</td><td>The length of the Mobile PIN Block. Valid values: '08' or '16'.</td></tr>
<tr><td>Mobile PIN Block</td><td>n B</td><td>The mobile PIN encrypted under the ZPK transport key</td></tr>
<tr><td>ZPK</td><td>'S' + n A</td><td>Zone PIN key encrypted under LMK key block and should comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'P0', '72'</td><td>'T', 'A'</td><td>'B', 'D', 'N'</td></tr>
</table></td></tr>
<tr><td>Mobile PIN Block Format Code</td><td>2 N</td><td>The PIN Block format of the supplied Mobile PIN.
Valid values are:
'34': Standard EMV 1996 format
'47': ISO 9564-1 PIN Block Format 3 (3DES)
'48': ISO 9564-1 PIN Block Format 4 (AES)</td></tr>
<tr><td>Account Number Length</td><td>2 N</td><td>Only present if Mobile PIN Block Format Code is '47' or '48'.
The number of digits in the following Account Number field.</td></tr>
<tr><td>Account Number</td><td>n N</td><td>Only present if Mobile PIN Block Format Code is '47' or '48'.
If Mobile PIN Block Format = '47':
The 12-19 digits of the Account Number, excluding the check digit.
If Mobile PIN Block Format = '48':
The 12-19 digits of the Account Number, including the check digit.</td></tr>
<tr><td>Session Key Method</td><td>1 N</td><td>Value '1': EMV CSK</td></tr>
<tr><td>Single Use Key Method</td><td>1 N</td><td>Value '1': XOR</td></tr>
<tr><td>SUK_Info</td><td>1 B</td><td>For example, 0x38 to denote release 1.0 for Mastercard Cloud Based payments</td></tr>
<tr><td>DC_CP Hash Code</td><td>32 B</td><td>SHA 256 hash code over the Digitised Card Profile data DC_CP. The DC_CP will have been created using the 'IU' Generate Secure Message function which returns the hash code over the plain text DC-CP data.</td></tr>
<tr><td>RFU Byte</td><td>1 B</td><td>Set to 0x00 as reserved for future use.</td></tr>
<tr><td>Template Length</td><td>4 N</td><td>The length of the DC_SUK Template.</td></tr>
<tr><td>DC_SUK Template</td><td>n A</td><td>The DC_SUK template in JSON format into which the generated keys and input data will be inserted.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value '%'. Optional; if present, the following field must be present.</td></tr>
<tr><td>LMK Identifier</td><td>2 N</td><td>LMK identifier; min value = '00'; max value is defined by licence; must be present if the above Delimiter is present.</td></tr>
<tr><td>End Message Delimiter</td><td>1 C</td><td>Must be present if a message trailer is present. Value X'19.</td></tr>
<tr><td>Message Trailer</td><td>n A</td><td>Optional. Maximum length 32 characters.</td></tr>
</table>

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'IZ'.
Error Code	2 A	'00': No error '06': Invalid YHHHHCC value '07': Invalid KCV type '20': Invalid PIN Block '23': Invalid PIN Block Format code '26': Invalid key scheme '27' : Key Scheme (KEK) mismatch 'A1': Invalid LMK scheme 'D4': MS_KEY_CONF or Output KEK – key block error 'D5': MS_KEY_MAC – key block error 'DA': KEK – key block error 'DB': ZPK – key block error 'DC': KEK parity error 'DD': DK parity error 'E0': Invalid Mode Flag 'E1': CMK_CL_UMD key parity error 'E2': CMK_RP_UMD key parity error 'E3': CMK_CL_MD key parity error 'E4': CMK_RP_MD key parity error 'E5': CMK_IDN parity error 'E6': Invalid Session Key Method 'E7': Invalid Single Use Key method 'E8': ATC will exceed maximum value 'E9': Invalid DC_SUK block count specified 'EA': Invalid JSON Template 'EB': Missing entry in template 'EC': Invalid RFU byte 'ED': Invalid DC_SUK_ID length 'EF': Invalid Key Generation Option value 'F1': Invalid session key length 'F2': Mobile data Encryption key block error 'F3': Invalid PIN Block Decryption Key Type 'F4': Invalid Block Cipher 'F5': Invalid Block Cipher Mode 'F6': Mobile data decryption key error Or a standard error code.
If Error Code = 'D4', 'D5', 'DA' or 'DB', the following field will be present:		
Additional Error Code	2 A	The key block specific error code.
Final ATC	2 B	The updated ATC value
DC_SUK Block Length	6 N	The length of the Encrypted DC_SUK Block to follow
Encrypted DC_SUK Block	n B	The DC_SUK Block will consist of 1 or more 116 byte DC_SUK containers AES encrypted using Counter Mode under the Remote Management MS_KEY_CONF session key.
Encrypted DC_SUK Block MAC	8 B	The MAC over the Encrypted DC_SUK Block using the MS_KEY_MAC session key and AES with MAC Algorithm 1 in ISO/IEC 9797-1 with padding method 2.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Generate Digitized Card Single Use Keys (Mode Flag '1': MCBP SUK Derivation; include PIN; output in MC SDK JSON format)

Variant LMK ☒	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Not required**	

Function: Derive Single Use Keys (SUKs) for Mastercard Cloud Based Payments (MCBP). The user and mobile PINs will be used during the production of the SUKs. The keys will be output in Mastercard's SDK JSON format.

Notes: This command generates one or more DC_SUK blocks that contain the single use keys for user and mobile device authentication for contactless and remote payments, the ATC and the Dynamic Number for use in CDA.

For details of the JSON format used in the DC_SUK block, refer to Appendix O – DC_SUK block template.

The DC_SUK blocks are concatenated together and encrypted using the Remote Management Session key for confidentiality MS_KEY_CONF and MAC protected using the Remote Management Session key for integrity MS_KEY_MAC. The output is in the format suitable to loading directly to the Mobile Payment Application.

The ATC value is updated by this command according to the number of DC_SUK blocks generated.

The following input data elements will have been generated by the data preparation system using the 'KI' Generate Card Unique Keys function encrypted under a transport key:

- CMK_CL_UMD
- CMK_RP_UMD
- CMK_CL_MD
- CMK_RP_MD
- CMK_IDN

The transport keys will have been imported and encrypted under the Local Master Key.

The Mobile PIN block will be created using the PIN Translation commands and transported encrypted under a ZPK.

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">COMMAND MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Subsequently returned to the Host unchanged.</td></tr>
<tr><td>Command Code</td><td>2 A</td><td>Value 'IY'.</td></tr>
<tr><td>Mode Flag</td><td>1 N</td><td>'1': MCBP SUK Derivation; include PIN; output in MC SDK JSON format</td></tr>
<tr><td>Key Generation Option Delimiter</td><td>1 A</td><td>Value '$'. Optional; if present, the following field must be present.</td></tr>
<tr><td>Key Generation Option</td><td>1 N</td><td>'0': Generate SUKs for Contactless Transactions (with PIN)
'1': Generate SUKs for Remote Payment Transactions (with PIN)
'2': Generate SUKs for Contactless Transactions (no PIN)
'3': Generate SUKs for Remote Payment Transactions (no PIN)
'4': Generate SUKs for Contactless & Remote Payment Transactions (no PIN)</td></tr>
<tr><td>Message Counter</td><td>5 N</td><td>The message counter is maintained by the CMS and is incremented for each message sent from the CMS to MPA.
Maximum value is 65535 or 0x00FFFF.</td></tr>
<tr><td>MS_KEY_CONF</td><td>'S' + n A</td><td>The Remote Management Session Key for message confidentiality, encrypted under AES Key Block LMK and which must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'35'</td><td>'A'</td><td>'B', 'D', 'E', 'N'</td></tr>
</table></td></tr>
<tr><td>MS_KEY_MAC</td><td>'S' + n A</td><td>The Remote Management Session Key for message integrity, encrypted under AES Key Block LMK and which must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'36'</td><td>'A'</td><td>'C', 'G', 'V', 'N'</td></tr>
</table></td></tr>
<tr><td>Start ATC</td><td>2 B</td><td>The start value for the Application Transaction Counter</td></tr>
<tr><td>Number of DC_SUK</td><td>2 N</td><td>The number of DC_SUK blocks to generate.
Value: '01'…'50'.</td></tr>
<tr><td>CMK_CL_UMD</td><td>'X' + 16 B</td><td>This field is NOT present when Key Generation Option = '1' or '3'.
User and Mobile Device Authentication Key for contactless transactions. 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>CMK_RP_UMD</td><td>'X' + 16 B</td><td>This field is NOT present when Key Generation Option = '0' or '2'.
User and Mobile Device Authentication Key for remote payments. 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>CMK_CL_MD</td><td>'X' + 16 B</td><td>This field is NOT present when Key Generation Option = '1' or '3'.
Mobile Device Authentication Key for contactless transactions. 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>CMK_RP_MD</td><td>'X' + 16 B</td><td>This field is NOT present when Key Generation Option = '0' or '2'.
User and Mobile Device Authentication Key for remote payments. 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>CMK_IDN</td><td>'X' + 16 B</td><td>CDA Dynamic Number generation key encrypted 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>Atalla Variant</td><td>2 N</td><td>Atalla variant; for use in systems with Atalla equipment.
Must be set to '00' if not used.</td></tr>
<tr><td>Transport KEK</td><td>'S' + n A</td><td>The transport KEK key encrypted under LMK key block and should comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'54'</td><td>'T', 'A'</td><td>'B', 'D', 'E', 'N'</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">If Key Generation Option = '2', '3' or '4', the following 6 fields are omitted:</td></tr>
<tr><td>Mobile PIN Block Length</td><td>2 N</td><td>The length of the Mobile PIN Block. Valid values: '08' or '16'.</td></tr>
<tr><td>Mobile PIN Block</td><td>n B</td><td>The mobile PIN encrypted under the ZPK transport key</td></tr>
<tr><td>ZPK</td><td>'S' + n A</td><td>Zone PIN key encrypted under LMK key block and should comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'P0', '72'</td><td>'T', 'A'</td><td>'B', 'D', 'N'</td></tr>
</table></td></tr>
<tr><td>Mobile PIN Block Format Code</td><td>2 N</td><td>The PIN Block format of the supplied Mobile PIN.
Valid values are:
'34': Standard EMV 1996 format
'47': ISO 9564-1 PIN Block Format 3 (3DES)
'48': ISO 9564-1 PIN Block Format 4 (AES)</td></tr>
<tr><td>Account Number Length</td><td>2 N</td><td>Only present if Mobile PIN Block Format Code is '47' or '48'.
The number of digits in the following Account Number field.</td></tr>
<tr><td>Account Number</td><td>n N</td><td>Only present if Mobile PIN Block Format Code is '47' or '48'.
The 12-19 digits of the Account Number, excluding the check digit.</td></tr>
<tr><td>Session Key Method</td><td>1 N</td><td>'1': EMV CSK</td></tr>
<tr><td>Single Use Key Method</td><td>1 N</td><td>'1': XOR</td></tr>
<tr><td>SUK_Info</td><td>1 B</td><td>For example, 0x38 to denote release 1.0 for Mastercard Cloud Based payments</td></tr>
<tr><td>DC_CP Hash Code</td><td>32 B</td><td>SHA 256 hash code over the Digitised Card Profile data DC_CP. The DC_CP will have been created using the 'IU' Generate Secure Message function which returns the hash code over the plain text DC-CP data.</td></tr>
<tr><td>RFU Byte</td><td>1 B</td><td>Set to 0x00 as reserved for future use.</td></tr>
<tr><td>Template Length</td><td>4 N</td><td>The length of the DC_SUK Template.</td></tr>
<tr><td>DC_SUK Template</td><td>n A</td><td>The DC_SUK template in JSON format into which the generated keys and input data will be inserted.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value '%'. Optional; if present, the following field must be present.</td></tr>
<tr><td>LMK Identifier</td><td>2 N</td><td>LMK identifier; min value = '00'; max value is defined by licence; must be present if the above Delimiter is present.</td></tr>
<tr><td>End Message Delimiter</td><td>1 C</td><td>Must be present if a message trailer is present. Value X'19.</td></tr>
<tr><td>Message Trailer</td><td>n A</td><td>Optional. Maximum length 32 characters.</td></tr>
</table>

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'IZ'.
Error Code	2 A	'00': No error '06': Invalid YHHHHCC value '07': Invalid KCV type '20': Invalid PIN Block '23': Invalid PIN Block Format code '26': Invalid key scheme '27' : Key Scheme (KEK) mismatch 'A1': Invalid LMK scheme 'D4': MS_KEY_CONF or Output KEK – key block error 'D5': MS_KEY_MAC – key block error 'DA': KEK – key block error 'DB': ZPK – key block error 'DC': KEK parity error 'DD': DK parity error 'E0': Invalid Mode Flag 'E1': CMK_CL_UMD key parity error 'E2': CMK_RP_UMD key parity error 'E3': CMK_CL_MD key parity error 'E4': CMK_RP_MD key parity error 'E5': CMK_IDN parity error 'E6': Invalid Session Key Method 'E7': Invalid Single Use Key method 'E8': ATC will exceed maximum value 'E9': Invalid DC_SUK block count specified 'EA': Invalid JSON Template 'EB': Missing entry in template 'EC': Invalid RFU byte 'ED': Invalid DC_SUK_ID length 'EF': Invalid Key Generation Option value 'F1': Invalid session key length 'F2': Mobile data Encryption key block error 'F3': Invalid PIN Block Decryption Key Type 'F4': Invalid Block Cipher 'F5': Invalid Block Cipher Mode 'F6': Mobile data decryption key error Or a standard error code.
If Error Code = 'D4', 'D5', 'DA' or 'DB', the following field will be present:		
Additional Error Code	2 A	The key block specific error code.
Final ATC	2 B	The updated ATC value
DC_SUK Block Length	6 N	The length of the Encrypted DC_SUK Block to follow
Encrypted DC_SUK Block	n B	The DC_SUK Block will consist of 1 or more 116 byte DC_SUK containers AES encrypted using Counter Mode under the Remote Management MS_KEY_CONF session key.
Encrypted DC_SUK Block MAC	8 B	The MAC over the Encrypted DC_SUK Block using the MS_KEY_MAC session key and AES with MAC Algorithm 1 in ISO/IEC 9797-1 with padding method 2.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Generate Digitized Card Single Use Keys (Mode Flag '2': VCP LUK from Card Key)

Variant LMK ☒	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Not required**	

Function: Derive Limited Use Keys (LUKs) for Visa Cloud-based Payments.

Notes: Allows a Visa LUK to be generated from a unique card key generated with a previous call to 'KI' with Key Derivation Method 'A' or 'B'. 'IY' should be used instead of 'KI' by systems that do not store the PAN.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'IY'.
Mode Flag	1 N	'2': VCP LUK from Card Key
DK (KEK)	1 A + 16 B	The derived unique key encrypted under the KEK.
YHHHHCC	7 N	The Year/Hour/Counter value used to derive the Limited Use Key (LUK) to produce the Limited Use Cryptogram (LUC) where: Y (0-9) : least significant digit of current year HHHH (0001-8784) : number of hours since Jan 1st of the current year CC (00-99) : counter that starts at 00 at the beginning of each hour and is incremented by 1 each time a Limited Use Key is generated.
Transport KEK	'S' + n A	The transport KEK key encrypted under LMK key block and should comply with: **Key Usage**: '54' | **Algorithm**: 'T' | **Mode of Use**: 'B', 'D', 'E', 'N'
Key Scheme	1 A	The Key scheme for encrypting the LUK key under KEK. Valid values are 'U' and 'X'. See the Key Scheme Table in Appendix A of the *payShield 9000 Host Programmer's Manual*.
Key Check Value Type	1 A	'0': KCV for entire key, 6 bytes long '1': KCV for entire key, 3 bytes long
Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by licence; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'IZ'.
Error Code	2 A	'00': No error '06': Invalid YHHHHCC value '07': Invalid KCV type '20': Invalid PIN Block '23': Invalid PIN Block Format code '26': Invalid key scheme '27' : Key Scheme (KEK) mismatch 'A1': Invalid LMK scheme 'D4': MS_KEY_CONF or Output KEK – key block error 'D5': MS_KEY_MAC – key block error 'DA': KEK – key block error 'DB': ZPK – key block error 'DC': KEK parity error 'DD': DK parity error 'E0': Invalid Mode Flag 'E1': CMK_CL_UMD key parity error 'E2': CMK_RP_UMD key parity error 'E3': CMK_CL_MD key parity error 'E4': CMK_RP_MD key parity error 'E5': CMK_IDN parity error 'E6': Invalid Session Key Method 'E7': Invalid Single Use Key method 'E8': ATC will exceed maximum value 'E9': Invalid DC_SUK block count specified 'EA': Invalid JSON Template 'EB': Missing entry in template 'EC': Invalid RFU byte 'ED': Invalid DC_SUK_ID length 'EF': Invalid Key Generation Option value 'F1': Invalid session key length 'F2': Mobile data Encryption key block error 'F3': Invalid PIN Block Decryption Key Type 'F4': Invalid Block Cipher 'F5': Invalid Block Cipher Mode 'F6': Mobile data decryption key error Or a standard error code.
If Error Code = 'D4', 'D5', 'DA' or 'DB', the following field will be present:		
Additional Error Code	2 A	The key block specific error code.
LUK (KEK)	1 A + 16 B	The Limited Use Key, encrypted under the KEK according to the Key Scheme.
Key Check Value	6 B or 3 B	The key check value of the Limited Use Key.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Generate Digitized Card Single Use Keys (Mode Flag '3': MCBP SK from MDES; add PIN; output encrypted SUK under KEK.)

Variant LMK ☒	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Not required**	

Function: Derive a Single Use Key (SUK) for Mastercard Cloud Based Payments (MCBP). The user and mobile PINs are used in the production of the SUK. The SUK is output encrypted under a supplied KEK.

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">COMMAND MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Subsequently returned to the Host unchanged.</td></tr>
<tr><td>Command Code</td><td>2 A</td><td>Value 'IY'.</td></tr>
<tr><td>Mode Flag</td><td>1 N</td><td>'3': MCBP SK from MDES; add PIN; output encrypted SUK under KEK</td></tr>
<tr><td>Session Key Length</td><td>2 N</td><td>The length of the Session Key in bytes.</td></tr>
<tr><td>Session Key</td><td>n B</td><td>The session key, encrypted under the transport key 'Mobile Data Decryption Key' using AES ECB</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value ';'.</td></tr>
<tr><td>Mobile Data Decryption Key</td><td>'S' + n A</td><td>The transport decryption key, encrypted under the LMK, and should comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'21'</td><td>'A'</td><td>'B', 'D', 'N'</td></tr>
</table></td></tr>
<tr><td>Mobile Data Encryption Key</td><td>'S' + n A</td><td>The transport encryption key, encrypted under the LMK, and should comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'21'</td><td>'A'</td><td>'B', 'E', 'N'</td></tr>
</table></td></tr>
<tr><td>Mobile PIN Block Decryption Key Type</td><td>1 N</td><td>Valid values are:
'0': Mobile PIN Block encrypted under the LMK
'1': Mobile PIN Block encrypted under ZPK</td></tr>
<tr><td>Mobile PIN Block Length</td><td>2 N</td><td>Only present if Mobile PIN Block Decryption Key Type='1'.
The length of the Mobile PIN Block.
Value: '08' or '16'.</td></tr>
<tr><td>Mobile PIN Block</td><td>n B</td><td>The encrypted Mobile PIN Block.
The Mobile PIN Block, encrypted under the supplied ZPK.</td></tr>
<tr><td>ZPK</td><td></td><td>Only present if Mobile PIN Block Decryption Key Type is '1'.
The Zone PIN Key, encrypted under the LMK.
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'P0', '72'</td><td>'T', 'A'</td><td>'B', 'D', 'N'</td></tr>
</table></td></tr>
<tr><td>Mobile PIN Block Format Code</td><td>2 N</td><td>Only present if Mobile PIN Block Decryption Key Type is '1'.
The PIN Block format of the supplied Mobile PIN.
Valid values are:
'34': Standard EMV 1996 format
'47': ISO 9564-1 PIN Block Format 3 (3DES)
'48': ISO 9564-1 PIN Block Format 4 (AES)</td></tr>
<tr><td>Account Number Length</td><td>2 N</td><td>Not present if Mobile PIN Block Format Code is '34'.
The number of digits in the following Account Number field.</td></tr>
<tr><td>Account Number</td><td>n N</td><td>Not present if Mobile PIN Block Format Code is '34'.
The 12-19 digits of the Account Number, excluding the check digit.</td></tr>
<tr><td>Single Use Key Method</td><td>1 N</td><td>The single use key generation method.
Valid values are:
Value: '1': XOR</td></tr>
<tr><td>Key Check Value Type</td><td>1 A</td><td>Valid values are:
'0': KCV for entire key, 6 bytes long.
'1': KCV for entire key, 3 bytes long.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value '%'. Optional; if present, the following field must be present.</td></tr>
<tr><td>LMK Identifier</td><td>2 N</td><td>LMK identifier; min value = '00'; max value is defined by licence; must be present if the above Delimiter is present.</td></tr>
<tr><td>End Message Delimiter</td><td>1 C</td><td>Must be present if a message trailer is present. Value</td></tr>
</table>

Field	Length & Type	Details
		X'19'.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'IZ'.
Error Code	2 A	'00': No error '06': Invalid YHHHHCC value '07': Invalid KCV type '20': Invalid PIN Block '23': Invalid PIN Block Format code '26': Invalid key scheme '27' : Key Scheme (KEK) mismatch 'A1': Invalid LMK scheme 'D4': MS_KEY_CONF or Output KEK – key block error 'D5': MS_KEY_MAC – key block error 'DA': KEK – key block error 'DB': ZPK – key block error 'DC': KEK parity error 'DD': DK parity error 'E0': Invalid Mode Flag 'E1': CMK_CL_UMD key parity error 'E2': CMK_RP_UMD key parity error 'E3': CMK_CL_MD key parity error 'E4': CMK_RP_MD key parity error 'E5': CMK_IDN parity error 'E6': Invalid Session Key Method 'E7': Invalid Single Use Key method 'E8': ATC will exceed maximum value 'E9': Invalid DC_SUK block count specified 'EA': Invalid JSON Template 'EB': Missing entry in template 'EC': Invalid RFU byte 'ED': Invalid DC_SUK_ID length 'EF': Invalid Key Generation Option value 'F1': Invalid session key length 'F2': Mobile data Encryption key block error 'F3': Invalid PIN Block Decryption Key Type 'F4': Invalid Block Cipher 'F5': Invalid Block Cipher Mode 'F6': Mobile data decryption key error Or a standard error code.
If Error Code = 'D4', 'D5', 'DA' or 'DB', the following field will be present:		
Additional Error Code	2 A	The key block specific error code.
Single Use Key Length	2 N	The length of the Single Use Key.
Single Use Key	n B	The single use key, encrypted under the Mobile Data Encryption Key transport key, using AES ECB.
Key Check Value	6 B or 3 B	The key check value of the Limited Use Key.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19'.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Generate Digitized Card Single Use Keys (Mode Flag '4': MCBP SUK Derivation; output encrypted under KEK)

Variant LMK ☒	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Not required**	

Function: Derive Single Use Keys (SUKs) for Mastercard Cloud-Based Payments (MCBP). No PIN is used in the production of the SUKs. The SUKs are output encrypted under a supplied KEK.

Notes: The following input data elements will have been generated by the data preparation system using the 'KI' Generate Card Unique Keys function encrypted under a transport key:

- CMK_CL_UMD
- CMK_RP_UMD
- CMK_CL_MD
- CMK_RP_MD
- CMK_IDN

The transport keys will have been imported and encrypted under the Local Master Key.

The Mobile PIN block will be created using the PIN Translation commands and transported encrypted under a ZPK.

The following keys are output, with their associated check value:

- SUK_CL_UMD
- SK_CL_MD
- SUK_RP_UMD
- SK_RP_MD

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">COMMAND MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Subsequently returned to the Host unchanged.</td></tr>
<tr><td>Command Code</td><td>2 A</td><td>Value 'IY'.</td></tr>
<tr><td>Mode Flag</td><td>1 N</td><td>'4': MCBP SUK Derivation; output encrypted under KEK</td></tr>
<tr><td>Key Generation Option Delimiter</td><td>1 A</td><td>Value '$'. Optional; if present, the following field must be present.</td></tr>
<tr><td>Key Generation Option</td><td>1 N</td><td>'0': Generate SUKs for Contactless Transactions (with PIN)
'1': Generate SUKs for Remote Payment Transactions (with PIN)
'2': Generate SUKs for Contactless Transactions (no PIN)
'3': Generate SUKs for Remote Payment Transactions (no PIN)
'4': Generate SUKs for Contactless & Remote Payment Transactions (no PIN)</td></tr>
<tr><td>Output KEK</td><td>'S' + n A</td><td>The Remote Management Session Key for message integrity, encrypted under AES Key Block LMK and which must comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'54','21'</td><td>'T', 'A'</td><td>'B', 'D', 'E', 'N'</td></tr></table></td></tr>
<tr><td>Start ATC</td><td>2 B</td><td>The start value for the Application Transaction Counter</td></tr>
<tr><td>Number of DC_SUK</td><td>2 N</td><td>The number of DC_SUK blocks to generate.
Value: '01' … '50'.</td></tr>
<tr><td>CMK_CL_UMD</td><td>'X' + 16 B</td><td>This field is NOT present when Key Generation Option is specified as '1' or '3'.
User and Mobile Device Authentication Key for contactless transactions. 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>CMK_RP_UMD</td><td>'X' + 16 B</td><td>This field is NOT present when Key Generation Option is specified as '0' or '2'.
User and Mobile Device Authentication Key for remote payments. 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>CMK_CL_MD</td><td>'X' + 16 B</td><td>This field is NOT present when Key Generation Option is specified as '1' or '3'.
Mobile Device Authentication Key for contactless transactions. 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>CMK_RP_MD</td><td>'X' + 16 B</td><td>This field is NOT present when Key Generation Option is specified as '0' or '2'.
User and Mobile Device Authentication Key for remote payments. 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>CMK_IDN</td><td>'X' + 16 B</td><td>CDA Dynamic Number generation key encrypted 3DES ECB Encrypted under the transport KEK.</td></tr>
<tr><td>Atalla Variant</td><td>2 N</td><td>Atalla variant; for use in systems with Atalla equipment.
Must be set to '00' if not used.</td></tr>
<tr><td>KEK</td><td>'S' + n A</td><td>The transport KEK key encrypted under LMK key block and should comply with:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'54'</td><td>'T', 'A'</td><td>'B', 'D', 'E', 'N'</td></tr></table></td></tr>
<tr><td colspan="3">If Key Generation Option is '2', '3' or '4', the following 6 fields are omitted:</td></tr>
<tr><td>Mobile PIN Block Length</td><td>2 N</td><td>The length of the Mobile PIN Block. Valid values: '08' or '16'.</td></tr>
<tr><td>Mobile PIN Block</td><td>n B</td><td>The mobile PIN encrypted under the ZPK transport key</td></tr>
<tr><td>ZPK</td><td>'S' + n A</td><td>Zone PIN key encrypted under LMK key block and should comply with:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'P0', '72'</td><td>'T', 'A'</td><td>'B', 'D', 'N'</td></tr></table></td></tr>
<tr><td>Mobile PIN Block</td><td>2 N</td><td>The PIN Block format of the supplied Mobile PIN.</td></tr>
</table>

Field	Length & Type	Details
Format Code		Valid values are: '34': Standard EMV 1996 format '47': ISO 9564-1 PIN Block Format 3 (3DES) '48': ISO 9564-1 PIN Block Format 4 (AES)
Account Number Length	2 N	Only present if Mobile PIN Block Format Code is '47' or '48'. The number of digits in the following Account Number field.
Account Number	n N	Only present if Mobile PIN Block Format Code is '47' or '48'. The 12-19 digits of the Account Number, excluding the check digit.
Session Key Method	1 N	Value: '1': EMV CSK
Single Use Key Method	1 N	Value: '1': XOR
Block Cipher	1 N	The block cipher for encrypting the output single use keys. Valid values are: '0': AES (Output KEK Algorithm must be 'A') '1': 3DES (Output KEK Algorithm must be 'T')
Block Cipher Mode	1 N	The block cipher mode used for encrypting the output single use keys. Valid values are: '0': CBC '1': ECB
IV	16 H or 32 H	The initialisation vector. Only present if Block Cipher Mode is '0'.
Key Check Value Type	1 A	Valid values are: '0': KCV for entire key, 6 bytes long '1': KCV for entire key, 3 bytes long
Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by licence; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'IZ'.
Error Code	2 A	'00': No error '06': Invalid YHHHHCC value '07': Invalid KCV type '20': Invalid PIN Block '23': Invalid PIN Block Format code '26': Invalid key scheme '27': Key Scheme (KEK) mismatch 'A1': Invalid LMK scheme 'D4': MS_KEY_CONF or Output KEK – key block error 'D5': MS_KEY_MAC – key block error 'DA': KEK – key block error 'DB': ZPK – key block error 'DC': KEK parity error 'DD': DK parity error 'E0': Invalid Mode Flag 'E1': CMK_CL_UMD key parity error 'E2': CMK_RP_UMD key parity error 'E3': CMK_CL_MD key parity error 'E4': CMK_RP_MD key parity error 'E5': CMK_IDN parity error 'E6': Invalid Session Key Method 'E7': Invalid Single Use Key method 'E8': ATC will exceed maximum value 'E9': Invalid DC_SUK block count specified 'EA': Invalid JSON Template 'EB': Missing entry in template 'EC': Invalid RFU byte 'ED': Invalid DC_SUK_ID length 'EF': Invalid Key Generation Option value 'F1': Invalid session key length 'F2': Mobile data Encryption key block error 'F3': Invalid PIN Block Decryption Key Type 'F4': Invalid Block Cipher 'F5': Invalid Block Cipher Mode 'F6': Mobile data decryption key error Or a standard error code.
If Error Code = 'D4', 'D5', 'DA' or 'DB', the following field will be present:		
Additional Error Code	2 A	The key block specific error code.
Each key will be encrypted under the 'Output KEK' using the 'Block Cipher'. If the optional 'Key Generation Option' delimiter was specified and the 'Key Generation Option' set to '0' or '2', the SUK_RP_UMD and SK_RP_MD keys with their KCVS will not be output. If the optional 'Key Generation Option' delimiter was specified and the 'Key Generation Option' set to '1' or '3', the SUK_CL_UMD and SK_CL_MD keys with their KCVS will not be output.		
ATC	4 H	Application Transaction Counter
Single Use Key Length	2 N	The length of the Single Use Keys
SUK_CL_UMD	n B	The Contactless Single Use Key for User and Mobile Device Authentication
SUK_CL_UMD KCV	6B or 3B	The key check value
SK_CL_MD	n B	The Contactless Single Use Key for Mobile Device Authentication
SK_CL_MD KCV	6B or 3B	The key check value
SUK_RP_UMD	n B	The Remote Payments Single Use Key for User and Mobile Device Authentication
SUK_RP_UMD KCV	6B or 3B	The key check value
SK_RP_MD	n B	The Remote Payment Single Use Key for Mobile Device Authentication
SK_RP_MD KCV	6B or 3B	The key check value
IDN	32 H	Dynamic Number in ASCII hexadecimal
Delimiter	1 A	';' Delimiter to denote end of keys generated using the

Field	Length & Type	Details
		ATC
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Generate Digitized Card Single Use Keys (Mode Flag '5': LUK from Card Key with PIN)

Variant LMK ☒	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Not required**	

Function: Derive a Limited Use Key (LUK) using a proprietary derivation method. The mobile PIN is used in the production of the LUK.

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">COMMAND MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Subsequently returned to the Host unchanged.</td></tr>
<tr><td>Command Code</td><td>2 A</td><td>Value 'IY'.</td></tr>
<tr><td>Mode Flag</td><td>1 N</td><td>'5': Proprietary unique session key with PIN</td></tr>
<tr><td>DK (KEK)</td><td>1 A + 16 B</td><td>The derived unique key encrypted under the KEK.</td></tr>
<tr><td>YHHHHCC</td><td>7 N</td><td>The Year/Hour/Counter value used to derive the Limited Use Key (LUK) to produce the Limited Use Cryptogram (LUC) where:
Y (0-9) : least significant digit of current year
HHHH (0001-8784) : number of hours since Jan 1st of the current year
CC (00-99) : counter that starts at 00 at the beginning of each hour and is incremented by 1 each time a Limited Use Key is generated.</td></tr>
<tr><td>KEK</td><td>'S' + n A</td><td>The transport KEK key encrypted under LMK key block and should comply with:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'54'</td><td>'T'</td><td>'B', 'D', 'E', 'N'</td></tr></table></td></tr>
<tr><td>Key Scheme</td><td>1 A</td><td>The Key scheme for encrypting the LUK key under KEK.
Valid values are 'U' and 'X'.
See the Key Scheme Table in Appendix A of the payShield 9000 Host Programmer's Manual.</td></tr>
<tr><td>Key Check Value Type</td><td>1 A</td><td>'0': KCV for entire key, 6 bytes long
'1': KCV for entire key, 3 bytes long</td></tr>
<tr><td>Single Use Key Method</td><td>1 N</td><td>Method by which the PIN is combined with the LUK
'1': XOR</td></tr>
<tr><td>Mobile PIN Block Length</td><td>2 N</td><td>The length of the Mobile PIN Block.</td></tr>
<tr><td>Mobile PIN Block</td><td>n B</td><td>The mobile PIN encrypted under the ZPK.</td></tr>
<tr><td>ZPK</td><td>'S' + n A</td><td>The Zone PIN Key, encrypted under the LMK and should comply with:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'P0', '72'</td><td>'T', 'A'</td><td>'B', 'D', 'N'</td></tr></table></td></tr>
<tr><td>Mobile PIN Block Format Code</td><td>2 N</td><td>The PIN Block format of the supplied Mobile PIN. Valid values are:
'34' - Standard EMV 1996 format
'47': ISO 9564-1 PIN Block Format 3 (3DES)
'48': ISO 9564-1 PIN Block Format 4 (AES)</td></tr>
<tr><td>Account Number Length</td><td>2 N</td><td>Only present if Mobile PIN Block Format Code is '47' or' 48'.
The number of digits in the following Account Number field.</td></tr>
<tr><td>Account Number</td><td>n N</td><td>Only present if Mobile PIN Block Format Code is '47' or' 48'.
The 12-19 digits of the Account Number, excluding the check digit.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value '%'. Optional; if present, the following field must be present.</td></tr>
<tr><td>LMK Identifier</td><td>2 N</td><td>LMK identifier; min value = '00'; max value is defined by licence; must be present if the above Delimiter is present.</td></tr>
<tr><td>End Message Delimiter</td><td>1 C</td><td>Must be present if a message trailer is present. Value X'19.</td></tr>
<tr><td>Message Trailer</td><td>n A</td><td>Optional. Maximum length 32 characters.</td></tr>
</table>

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'IZ'.
Error Code	2 A	'00': No error '06': Invalid YHHHHCC value '07': Invalid KCV type '20': Invalid PIN Block '23': Invalid PIN Block Format code '26': Invalid key scheme '27' : Key Scheme (KEK) mismatch 'A1': Invalid LMK scheme 'D4': MS_KEY_CONF or Output KEK – key block error 'D5': MS_KEY_MAC – key block error 'DA': KEK – key block error 'DB': ZPK – key block error 'DC': KEK parity error 'DD': DK parity error 'E0': Invalid Mode Flag 'E1': CMK_CL_UMD key parity error 'E2': CMK_RP_UMD key parity error 'E3': CMK_CL_MD key parity error 'E4': CMK_RP_MD key parity error 'E5': CMK_IDN parity error 'E6': Invalid Session Key Method 'E7': Invalid Single Use Key method 'E8': ATC will exceed maximum value 'E9': Invalid DC_SUK block count specified 'EA': Invalid JSON Template 'EB': Missing entry in template 'EC': Invalid RFU byte 'ED': Invalid DC_SUK_ID length 'EF': Invalid Key Generation Option value 'F1': Invalid session key length 'F2': Mobile data Encryption key block error 'F3': Invalid PIN Block Decryption Key Type 'F4': Invalid Block Cipher 'F5': Invalid Block Cipher Mode 'F6': Mobile data decryption key error Or a standard error code.
If Error Code = 'D4', 'D5', 'DA' or 'DB', the following field will be present:		
Additional Error Code	2 A	The key block specific error code.
LUK (KEK)	1 A + 16 B	The Limited Use Key, encrypted under the KEK according to the Key Scheme.
Key Check Value	6 B or 3 B	The key check value of the Limited Use Key.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Generate Remote Management Session ID and Session Keys

Variant LMK ☒	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Not required**	

Function: Generate Remote Management Session ID and Session Keys

Notes: The random Session ID is generated by the host Credentials Management System (CMS) to uniquely identify a remote management session between the Mobile Payment Application (MPA) and the host CMS.

The Remote Management Session keys are derived from the M_KEY_CONF and M_KEY_MAC keys and Session ID.

The M_KEY_CONF and M_KEY_MAC are generated using the 'A0' Generate Key command.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'IO'.
Control Bits	1 B	Mastercard defined version control and options. For example, set to 0x80 for Release 1.0 for Mastercard Cloud-Based Payments.
Expiry Date	6 N	Expiry date for the session, in the format 'YYMMDD', where: YY (00-99): least significant 2 digits of the year of the expiry date; MM (01-12): month of expiry date; DD (01-31): day of expiry date; Note this value is not validated by this command.
Remote Management Info	6 H	Defined by Mastercard. For example: '810000': when provisioning card profile DC_CP '820000': when provisioning key profile DC_SUK
M_KEY_CONF	'S' + n A	The Remote Management Master Key for message confidentiality, encrypted under AES Key Block LMK and which must comply with the following: **Key Usage**: '33' **Algorithm**: 'A' **Mode of Use**: 'X', 'N'
M_KEY_MAC	'S' + n A	The Remote Management Master Key for message integrity, encrypted under AES Key Block LMK and which must comply with the following: **Key Usage**: '34' **Algorithm**: 'A' **Mode of Use**: 'X', 'N'
Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by licence; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">RESPONSE MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Returned to the Host unchanged.</td></tr>
<tr><td>Response Code</td><td>2 A</td><td>Value 'IP'.</td></tr>
<tr><td>Error Code</td><td>2 A</td><td>'00': No error
'15': Expiry Date not numeric
'A1': Invalid LMK scheme
'D1': M_KEY_CONF – key block error
'D2': M_KEY_MAC – key block error
Or a standard error code</td></tr>
<tr><td colspan="3">If Error Code = 'D1' or 'D2', the following field will be present:</td></tr>
<tr><td>Additional Error Code</td><td>2 A</td><td>The key block specific error code. See reference 1.</td></tr>
<tr><td colspan="3">If Error Code = '00', the following 5 fields will be present:</td></tr>
<tr><td>Encrypted Session ID under LMK</td><td>32 B</td><td>The Session ID encrypted under the LMK for local use.</td></tr>
<tr><td>Encrypted Session ID under M_KEY_CONF</td><td>32 B</td><td>The Session ID encrypted under M_KEY_CONF using AES and ECB mode for transfer to MPA.</td></tr>
<tr><td>Session ID MAC</td><td>8 B</td><td>MAC over Session ID using M_KEY_MAC and AES with MAC Algorithm 1 in ISO/IEC 9797-1, padding method 2.</td></tr>
<tr><td>MS_KEY_CONF</td><td>'S' + n A</td><td>The Remote Management Session Key for message confidentiality, encrypted under the AES Key Block LMK.
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'35'</td><td>'A'</td><td>'B'</td></tr>
</table></td></tr>
<tr><td>MS_KEY_MAC</td><td>'S' + n A</td><td>The Remote Management Session Key for message integrity, encrypted under the AES Key Block LMK.
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'36'</td><td>'A'</td><td>'C'</td></tr>
</table></td></tr>
<tr><td>End Message Delimiter</td><td>1 C</td><td>Present only if present in the command message. Value X'19.</td></tr>
<tr><td>Message Trailer</td><td>n A</td><td>Present only if present in the command message. Maximum length 32 characters.</td></tr>
</table>

Contactless Cards Data Preparation Commands

Use of this command requires the optional Contactless Cards Data Preparation license.

The HSM provides the following host command to support generation of the data required for issuing magnetic stripe based Contactless Cards:

Command	Page
Generate IVCVC3 and Static CVC3 (NY)	56

Generate IVCVC3 and Static CVC3

Variant LMK ☑	Key Block LMK ☑
License: **HSM9-LIC011**	
Authorization: **Not required**	

Function: Generate IVCVC3 & CVC3 or PINIVCVC3 & PINCVC3.

Notes: Mastercard's PayPass requires an IVCVC3 for card personalization. Calculation of the Static CVC3 requires the IVCVC3 which is a MAC calculated over the static part of $Track_1$ or $Track_2$ data using the *DK-CVC3*. This command creates the IVCVC3 and the CVC3 from the Track (1 or 2) Data provided.

Mobile PayPass requires the use of a PINIVCVC3 value for the calculation of the PINCVC3.

From the Mastercard Mobile PayPass M/Chip specifications:

The PINIVCVC3 is an issuer proprietary static data object that is used as input for the generation of the CVC 3 cryptogram when the reader supports the Mobile extensions and when the PIN has been successfully verified offline.

PIN initialization vector (PINIVCVC3) = initialization vector (IVCVC3) XOR '9559'

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'NY'.
Scheme ID	1 N	Identifier for Card Scheme: '1': Mastercard PayPass (IVCVC3 and CVC3) '2': Mastercard PayPass (PINIVCVC3 and PINCVC3) Other values are RFU.
MK-CVC3		The Issuer Master Key for calculating the CVC3.
	32 H or 'U' + 32 H	For a Variant LMK, the MK-CVC3 is encrypted under LMK 28-29/7.
	or 'S' + n A	For a Key Block LMK, the MK-CVC3 must comply with: **Key Usage**: 'E0', 'E6', '32' / **Algorithm**: 'T' / **Mode of Use**: 'N', 'X'
Key Derivation Method	1 A	Method to be used to derive a unique key from the Master Key: 'A': EMV 4.1 Book 2 Option A method 'B': EMV 4.1 Book 2 Option B method
Derivation Data	n N	Concatenation of the Primary Account Number and 2 digit Sequence Number for the card. If the Sequence Number is not available it should be specified as '00'.
Delimiter	1 A	Delimiter. Value ';'.
Track Data Length	3 N	The length of the following field.
Track Data	n B	Static Track (1 or 2) Data.

Field	Length & Type	Details
Delimiter	1 A	Delimiter. Value ';'.
Unpredictable Number	8 N	Random number provided to the card by the terminal during a PayPass transaction.
ATC	5 N	Decimal value of Application Transaction Counter. Max value 65535 (2 byte field).
Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'NZ'.
Error Code	2 N	'00': No error '04': Unrecognized Key Derivation Method '05': Invalid Scheme ID '10': MK-CVC3 Parity Error 'EA': MK-CVC3 – key block specific error or a standard error code.
If Error Code = 'EA', the following field will be present:		
Additional Error Code	2 A	The key block specific error code
IVCVC3/PINIVCVC3	5 N	The calculated IVCVC3 or PINIVCVC3.
Static CVC3/PINCVC3	5 N	The calculated CVC3 or PINCVC3. In cases where only 3 or 4 digits are required, the application can truncate the returned value.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

EMV-based Cards Data Preparation Commands

Use of these commands requires the optional EMV-based Cards Data Preparation license.

The HSM provides the following host commands to support generation of the data required for basic EMV-based cards:

Generate Issuer RSA Key Set and Public Key Certificate

Variant LMK ☑	Key Block LMK ☑
License: **HSM9-LIC002** & **HSM9-LIC016**	
Authorization: **Required** Activity: **generate.rsa.host**	

Function: Generate an Issuer RSA Key Set and return the Public Key in the form of a Self-Signed Issuer Public Key Certificate following the format of the card association specified in the command.

Alternatively, generate a Self-Signed Issuer Public Key Certificate following the format of the card association specified in the command using a Private/Public key pair previously generated by this command. The command will verify that the Private/Public key pair correspond to each other.

Notes: The RSA Key Type for the Private Key will be set to 2 (Signature and Key Management). A Public Exponent of 65537 (2^{16} + 1) will be used unless another is specified. If a Public Exponent is supplied, it must be 3 or 65537; otherwise, an error will be returned by the HSM and no processing will take place.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'KE'.
Scheme ID	1 N	'0': Visa VSDC '1': Mastercard '2': American Express AEIPS V4.1 '3': JCB '4': Union Pay '5': Discover
Mode Flag	1 N	'0': Generate Issuer Key Pair and certificate, using Strong Primes '1': Generate certificate only with provided Issuer Key Pair '2': Generate Issuer Key Pair and certificate using Standard Primes
Hash Identifier	2 N	Identifier of algorithm used to hash data: '01': SHA-1
Signature Identifier	2 N	Identifier of signature algorithm: '01': RSA
If Mode Flag = '0' or '2', the following 5 fields must be present as noted below:		
Key Length	4 N	Modulus length in bits (must be a multiple of 8) minimum value = 0400, maximum value = 2040.
Authentication Data	n A	Optional. Additional data to be included in the Public Key MAC calculation (must not include ';').
Delimiter	1 A	Mandatory Delimiter. Value ';'.
Public Exponent Length	4 N	Optional. Length, in bits, of the Public Exponent. Must be supplied if Public Exponent is present in command message.
Public Exponent	n B	Optional. If supplied, it must follow guidelines described in the command notes. If not supplied then a default exponent of 65537 will be used.

<table>
<tr><td colspan="3">If Mode Flag = '1', the following 4 fields must be present as noted below:</td></tr>
<tr><td rowspan="4">Issuer Private Key Length</td><td></td><td>Length of the Issuer Private Key.</td></tr>
<tr><td>4 N</td><td>For a Variant LMK, the length, in bytes, of the Issuer Private Key field.</td></tr>
<tr><td>or
4 H</td><td>For a Key Block LMK, this field is ignored and should be set to 'FFFF'.</td></tr>
<tr><td></td><td></td></tr>
<tr><td rowspan="3">Issuer Private Key</td><td></td><td>The Issuer Private Key, encrypted under the LMK.</td></tr>
<tr><td>n B</td><td>For a Variant LMK, the Issuer Private Key is encrypted under LMK 34-35.</td></tr>
<tr><td>or
'S' + n B</td><td>For a Key Block LMK, the Issuer Private Key must comply with:
<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'03'</td><td>'R'</td><td>'S', 'D', 'N'</td></tr></table></td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Mandatory Delimiter. Value ';'</td></tr>
<tr><td>Issuer Public Key</td><td>n B</td><td>Issuer Public Key, DER encoded (unsigned) in ASN.1 format (sequence of modulus and exponent)</td></tr>
<tr><td colspan="3">The following fields must be present as noted below:</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Mandatory Delimiter. Value ';'.</td></tr>
<tr><td colspan="3">For Scheme-ID = '0', '1', '2', '3' and '4', the following 2 fields must be present:</td></tr>
<tr><td>Issuer Identifier (BIN)</td><td>8 H</td><td>Leftmost 3-8 digits from the Primary Account Number (PAN) padded to the right with Hex 'F's.</td></tr>
<tr><td>Certificate Expiration Date</td><td>4 N</td><td>Expiration date of certificate in Month and Year (MMYY) format.</td></tr>
<tr><td colspan="3">For Scheme-ID = '0' (Visa VSDC), the following 2 fields providing the data to be included in the self-signed certificate must be present. (See Appendix A – Self-Signed Issuer Public Key Certificate Format (Visa)).</td></tr>
<tr><td>Service Identifier</td><td>8 H</td><td>Identifies the specific Visa service padded to the right with Hex '0's.</td></tr>
<tr><td>Tracking Number</td><td>6 N</td><td>Certificate tracking number assigned by Visa.</td></tr>
<tr><td colspan="3">For Scheme-ID = '1' (Mastercard), the following 2 fields providing the data to be included in the self-signed certificate must be present (See Appendix B – Self-Signed Issuer Public Key Certificate Format (Mastercard)).</td></tr>
<tr><td>Certificate Serial Number</td><td>6 H</td><td>Certificate tracking number.</td></tr>
<tr><td>Issuer Public Key Index</td><td>6 H</td><td>Issuer assigned unique public key identifier.</td></tr>
<tr><td colspan="3">For Scheme-ID = '2' (American Express AEIPS V4.1), the following 2 fields providing the data to be included in the self-signed certificate must be present (See Appendix C – Self-Signed Issuer Public Key Certificate Format (American Express)):</td></tr>
<tr><td>Service Identifier</td><td>8 H</td><td>Identifies the specific American Express Product Identifier.</td></tr>
<tr><td>Tracking Number</td><td>6 N</td><td>Transmittal tracking number.</td></tr>
<tr><td colspan="3">For Scheme-ID = '3' (JCB), the following 2 fields providing the data to be included in the self-signed certificate must be present:</td></tr>
<tr><td>Certificate Serial Number</td><td>6 H</td><td>Certificate tracking number.</td></tr>
<tr><td>Issuer Public Key Index</td><td>6 H</td><td>Issuer assigned unique public key identifier.</td></tr>
<tr><td colspan="3">For Scheme-ID = '4' (Union Pay), the following 2 fields providing the data to be included in the self-signed certificate must be present:</td></tr>
<tr><td>Service Identifier</td><td>8 H</td><td>Identifies the specific Union Pay service padded to the right with Hex '0's.</td></tr>
<tr><td>Tracking Number</td><td>6 N</td><td>Application record No. in the issuer public key</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value '%'. Optional; if present, the following field must be present.</td></tr>
<tr><td>LMK Identifier</td><td>2 N</td><td>LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.</td></tr>
</table>

Field	Length & Type	Details
If Key Block LMK, the following fields must be present:		
Delimiter	1 A	Value '&'. Optional; can only be present when the generated private key is to be exported; if present, the following field must be present.
Modified Export Value	1 A	Character to be placed in the exportability field (byte 11) of the exported private key; only permitted values are 'N', 'S'; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'KF'.
Error Code	2 N	'00': No error '05': Invalid Scheme ID or Mode Flag '06': Invalid hash or signature identifier or a standard error code. For Mode = '0', '2' or '4' only: '03': Key Length error '07': Public exponent length error '08': Supplied public exponent value is not allowed For Mode = '1' or '3' only: '02': Public Key does not conform to encoding rules '09': Invalid Public key / Private key pair 'E8': Private Key – key block error 'E9': Public Key – key block error
If Error Code = 'E8' or 'E9', the following field will be present:		
Additional Error Code	2 A	The key block specific error code
If Mode Flag = '0' or '2', the following 4 fields will be present:		
MAC	4 B	Only present if using a Variant LMK. MAC on Issuer Public Key and Authentication Data calculated using LMK 36-37.
Issuer Public Key		The Issuer Public Key.
	n B	For a Variant LMK, the Issuer Public Key, DER encoded (unsigned) in ASN.1 format (sequence of modulus and exponent).
	or 'S' + n B	For a Key Block LMK, the Issuer Public Key will conform to: **Key Usage** / **Algorithm** / **Mode of Use** '02' / 'R' / 'V'
Issuer Private Key Length	4 N	Only present if using a Variant LMK. Length, in bytes, of the Issuer Private Key field.
Issuer Private Key		The Issuer Private Key.
	n B	For a Variant LMK, the Issuer Private Key is encrypted under LMK 34-35.
	or 'S' + n B	For a Key Block LMK, the Issuer Private Key will conform to: **Key Usage** / **Algorithm** / **Mode of Use** '03' / 'R' / 'S'

Field	Length & Type	Details
For Scheme-ID = '0', '1', '2', '3' or '4', the following fields will be present:		
Certificate Length	4 N	Length, in bytes, of Self-Signed Certificate.
Self-Signed Issuer Public Key Certificate	n B	Self-Signed Issuer Public Key Certificate (the concatenation of the Clear Data and the Self-Signed Certificate). See Appendix A, B or C depending on Scheme ID.
Hash Length	2 N	Length in hex characters of hash results in next field. This length will depend on the hash algorithm specified in the command message. For SHA-1, this length will be '40'.
Hash Value	n H	Hash value required for transfer of self-signed Issuer Public Key data as defined by Scheme ID.
For Scheme-ID = '5', the following field will be present:		
Issuer Public Key Modulus	n B	The Issuer Public Key Modulus value.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Validate an Issuer Public Key Certificate

Variant LMK ☑	Key Block LMK ☑
License: **HSM9-LIC002** & **HSM9-LIC016**	
Authorization: **Not required**	

Function:	Validate an Issuer public key certificate returned by the Certificate Authority and return the Public Key with its associated MAC. Optionally, verify the Public Key in the certificate corresponds to the Private Key for the key pair.
Notes:	A complete Issuer Public Key Certificate comprises an Unsigned Data section, an Issuer Public Key Certificate and, for American Express, Visa and Union Pay certificates, an optional Detached Signature. The Detached Signature is a signature on the first two parts of the certificate and will be validated by this command if present in the provided certificate.
Compatibility Note:	In this revision, the command has been enhanced to also return the Hash in the Detached Signature, if present in the Public Key Certificate, for American Express and Visa certificates.
Issuer Public Key Note:	• If the input Issuer Private Key's Mode of Use field = "S", then the output Issuer Public Key's Mode of Use field should be set to "V". • If the input Issuer Private Key's Mode of Use field = "N", then the output Issuer Public Key's Mode of Use field should be set to "N".

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">COMMAND MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Subsequently returned to the Host unchanged.</td></tr>
<tr><td>Command Code</td><td>2 A</td><td>Value 'KG'.</td></tr>
<tr><td>Scheme ID</td><td>1 N</td><td>'0': Visa VSDC
'1': Mastercard
'2': American Express AEIPS V4.1
'3': JCB
'4': Union Pay
'5': Discover</td></tr>
<tr><td>CA Public Key MAC</td><td>4 B</td><td>Only present if using a Variant LMK.
MAC on Public Key and Authentication Data calculated using LMK 36-37.</td></tr>
<tr><td rowspan="3">CA Public Key</td><td></td><td>CA Public key, DER encoded (unsigned) in ASN.1 format (Sequence of modulus, exponent).</td></tr>
<tr><td>n B</td><td>For a Variant LMK, the CA Public Key, DER encoded (unsigned) in ASN.1 format (sequence of modulus and exponent).</td></tr>
<tr><td>or
'S' + n B</td><td>For a Key Block LMK, the CA Public Key must comply with:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'02'</td><td>'R'</td><td>'N', 'V'</td></tr></table></td></tr>
<tr><td>CA Authentication Data</td><td>n A</td><td>Only present if using a Variant LMK.
Optional; additional data to be included in the CA public key MAC calculation (must not include ';').</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Mandatory Delimiter. Value ';'.</td></tr>
<tr><td>Certificate Length</td><td>4 N</td><td>Length, in bytes, of the Issuer Certificate.</td></tr>
<tr><td>Issuer Certificate</td><td>n B</td><td>Issuer Certificate, comprising of the Unsigned Data and the Issuer Public Key Certificate. See Appendix D, E, or F depending on Scheme ID.
For Scheme ID = '0' (Visa VSDC) and Scheme ID = '2' (American Express AEIPS V4.1), the Issuer certificate may also include a Detached Signature.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Mandatory Delimiter. Value ';'</td></tr>
<tr><td>Issuer Authentication Data</td><td>n A</td><td>Optional. Additional data to be included in the Issuer Public Key MAC calculation (must not include ';').</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Mandatory Delimiter. Value ';'</td></tr>
<tr><td rowspan="3">Issuer Private Key Length</td><td></td><td>Length of the Issuer Private Key.</td></tr>
<tr><td>4 N
or</td><td>For a Variant LMK, the length, in bytes, of the Issuer Private Key field.</td></tr>
<tr><td>4 H</td><td>For a Key Block LMK, this field is ignored and should be set to 'FFFF'.</td></tr>
<tr><td rowspan="3">Issuer Private Key</td><td></td><td>The Issuer Private Key.</td></tr>
<tr><td>n B
or</td><td>For a Variant LMK, the Issuer Private Key is encrypted under LMK 34-35.</td></tr>
<tr><td>'S' + n B</td><td>For a Key Block LMK, the Issuer Private Key must comply with:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'03'</td><td>'R'</td><td>'S', 'D', 'N'</td></tr></table></td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Mandatory Delimiter. Value ';'.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value '%'. Optional; if present, the following field must be present.</td></tr>
<tr><td>LMK Identifier</td><td>2 N</td><td>LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.</td></tr>
<tr><td>End Message Delimiter</td><td>1 C</td><td>Must be present if a message trailer is present. Value X'19.</td></tr>
<tr><td>Message Trailer</td><td>n A</td><td>Optional. Maximum length 32 characters.</td></tr>
</table>

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">RESPONSE MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Returned to the Host unchanged.</td></tr>
<tr><td>Response Code</td><td>2 A</td><td>Value 'KH'.</td></tr>
<tr><td>Error Code</td><td>2 N</td><td>'00': No error
'01': MAC verification failure
'05': Invalid Scheme ID
'06': Invalid Hash or Signature algorithm
'07': Certificate Hash validation failure
'08': CA PK does not conform to encoding rules
'09': Invalid Public key / Private key pair
'76': Public key length error
'78': Private key length error
'80': Certificate length error
'81': Invalid Certificate Format (Header or Trailer)
'E8': Invalid Private Key Block
'E9': Invalid Certificate Format Identifier
or a standard error code.
For Scheme ID = '0', '2' or '4' only:
'52': Invalid Certificate Extension Header
'53': Detached Signature format error
'54': Detached Signature length error
'55': Detached Signature error</td></tr>
<tr><td colspan="3">If Error Code = 'E8' or 'E9', the following field will be present:</td></tr>
<tr><td>Additional Error Code</td><td>2 A</td><td>The key block specific error code</td></tr>
<tr><td>MAC</td><td>4 B</td><td>Only present if using a Variant LMK.
MAC on Issuer Public Key and Authentication Data calculated using LMK 36-37.</td></tr>
<tr><td>Issuer Public Key</td><td></td><td>The Issuer Public key.</td></tr>
<tr><td></td><td>n B</td><td>For a Variant LMK, the Issuer Public Key, DER encoded (unsigned) in ASN.1 format (sequence of modulus and exponent).</td></tr>
<tr><td></td><td>or
'S' + n B</td><td>For a Key Block LMK, the Issuer Public Key will conform to:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'02'</td><td>'R'</td><td>'V'</td></tr>
</table></td></tr>
<tr><td>Hash Length</td><td>2 N</td><td>Length in hex characters of hash results in next field. This length will depend on the hash algorithm specified in the certificate. For SHA-1 this length will be 40.</td></tr>
<tr><td>Hash Value</td><td>n H</td><td>Hash Value from Issuer Public Key Certificate.</td></tr>
<tr><td>Issuer Identification Number</td><td>8 H</td><td>Issuer Identifier recovered from Issuer Public Key Certificate.</td></tr>
<tr><td>Certificate Expiry Date</td><td>4 N</td><td>Certificate Expiration Date (MMYY) recovered from the Issuer Public Key Certificate.</td></tr>
<tr><td>Certificate Serial/Tracking Number</td><td>6 H</td><td>Certificate Serial/Tracking Number recovered from the Issuer Public Key Certificate.</td></tr>
<tr><td colspan="3">For Scheme 0 (Visa VSDC) and 2 (American Express AEIPS V4.1) and '4' (Union Pay), the following fields will be present. Refer to Appendix D or F respectively for details on Detached Signatures.</td></tr>
<tr><td>Detached Signature Hash Length</td><td>2 N</td><td>Length in hex characters of hash results in next field. This length will depend on the hash algorithm specified in the detached signature. For SHA-1, this length will be '40'.
If a Detached Signature is not present in the Certificate, the length field will be '00' and the next field will not be present.</td></tr>
<tr><td>Detached Signature Hash Value</td><td>n H</td><td>Hash value in Detached Signature.</td></tr>
<tr><td>End Message Delimiter</td><td>1 C</td><td>Present only if present in the command message. Value X'19.</td></tr>
<tr><td>Message Trailer</td><td>n A</td><td>Present only if present in the command message. Maximum length 32 characters.</td></tr>
</table>

Generate Static Data Authentication Signature

Variant LMK ☑	Key Block LMK ☑
License: **HSM9-LIC002** & **HSM9-LIC016**	
Authorization: **Not required**	

Function: Sign Card Data using Issuer's Private Key

Notes: Automatic DAC generation is provided as an option (used by Mastercard schemes).

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'KM'.
Hash Identifier	2 N	Identifier of algorithm used to hash data. '01': SHA-1
Data Authentication Code	2 B	Data Authentication Code. A value must always be supplied but it will be ignored if the optional parameters at the end are supplied in which case the DAC is calculated.
Data Length	4 N	Length of Static Authentication Data field.
Static Authentication Data	n B	Static authentication data.
Delimiter	1 A	Mandatory Delimiter. Value ';'.
Issuer Private Key Flag	2 N	Flag to indicate location of the Issuer's Private key: If flag = '99' use Issuer Private key provided with command else flag = index of stored Issuer Private key.
Issuer Private Key Length		Length of the Issuer Private Key. Only present if Issuer Private Key Flag = '99'.
	4 N	For a Variant LMK, the length, in bytes, of the Issuer Private Key field.
	or 4 H	For a Key Block LMK, this field is ignored and should be set to 'FFFF'.
Issuer Private Key		The Issuer Private Key. Only present if Private Key Flag = '99'.
	n B	For a Variant LMK, the Issuer Private Key is encrypted under LMK 34-35.
	or 'S' + n B	For a Key Block LMK, the Issuer Private Key must comply with: **Key Usage**: '03' / **Algorithm**: 'R' / **Mode of Use**: 'S', 'N'
Delimiter	1 A	Optional Delimiter. Value ';'. Indicates the presence of the following 2 fields which allow a DAC to be calculated.
If above Delimiter is present, the following 2 fields must be present:		
MKDAC		Issuer Master Key for Data Authentication Code.
	32 H or 'U' + 32 H	For a Variant LMK, the MKDAC is encrypted under LMK 28-29/4.
	or 'S' + n A	For a Key Block LMK, the MKDAC must comply with: **Key Usage**: 'E3' / **Algorithm**: 'T' / **Mode of Use**: 'B', 'N'

Field	Length & Type	Details
PAN/PSN	16 N	Concatenation of the rightmost 14 digits of the Primary Account Number and 2 digit Sequence Number for the card. Pad on left with zeroes if required. If the Sequence Number is not available it should be specified as '00'.
Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'KN'.
Error Code	2 N	'00': No error '04': Invalid Private key flag '06': Invalid hash identifier '10': Parity error on MK DAC 'E8': Invalid Private key block 'E9': Invalid MK-DAC key block or a standard error code.
If Error Code = 'E8' or 'E9', the following field will be present:		
Additional Error Code	2 A	The key block specific error code
Signature Length	4 N	Length, in bytes, of the SDA signature.
Signature	n B	Calculated SDA signature.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Generate Card RSA Key Set and Public Key Certificate

Variant LMK ☑	Key Block LMK ☑
License: **HSM9-LIC002** & **HSM9-LIC016**	
Authorization: **Not required**	

Function: Generate a Card RSA Key Set and create the requested Certificate for the Public Key signed by the Issuer's Private Key.

Alternatively, create the requested certificate using the provided Public Key, reformatting and exporting the associated Private Key as requested. The command will verify that the provided Public Key corresponds to the Private Key for the key pair. The Key Type of the Private Key provided must be set to 3 (ICC Key).

Notes: A Public Exponent of 65537 (2^{16} + 1) will be used unless another is specified. If a Public Exponent is supplied, it must be 3 or 65537; otherwise, an error will be returned by the HSM and no processing will take place. See Appendix H – Private Key Encodings for discussion on alternative Chinese Remainder Theorem output formats.

Compatibility Note: In this revision, Mode 1 has been changed and it **will not** be backwards compatible. Originally, this mode only required the Card Public Key and it now requires both the Card Public and Private keys so that the Private Key can be formatted as required by card personalization systems. Previously, there was no functionality in the HSM to provide the Card Private Key in the required format. This has now been addressed. In addition, the provided Card Public key will no longer require a MAC since the command will verify that the key corresponds to the provided Private Key.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'KO'.
Scheme ID	1 N	'0': Visa VSDC '1': Mastercard '2': American Express AEIPS V4.1 '3': JCB '4': Union Pay '5': Discover *The Scheme ID parameter is not used in the processing at this time. It is included for any future differentiations in the Schemes.*
Certificate Type	1 N	'0': Card Public Key Certificate '1': Card PIN Encipherment Public Key Certificate
Mode Flag	1 N	'0': Generate RSA Key Set and Certificate, using Strong Primes '1': Generate Certificate with provided Public Key and export the provided Private Key under the KEK in the requested format '2': Generate RSA Key Set and Certificate, using Standard Primes
RSA Key Mode Flag	1 N	Indicates the type of keys required

		'0': Keys with q > p '2': Keys with p > q *Note: For provided key sets, the command will verify the type of key provided and modify the key parameters to comply with this condition.*
If Mode Flag = '0' or '2', the following field must be present:		
Key Length	4 N	Modulus length in bits (must be a multiple of 8) minimum value = '0400', maximum value = '2040'.
The following fields must be present as noted below:		
Card Private Key Output format	2 N	Output format for Card Private Key: '03': Output in the form of 5 Chinese Remainder Theorem components CBC encrypted under the KEK. See Appendix H – Private Key Encodings '04': Output the private key exponent (d) and modulus (n) encrypted under the KEK. See Appendix H – Private Key Encodings for format description. '05': Output the private key in the form of 5 CRT components encrypted under the KEK (format 03) and also in private key exponent (d) and modulus (n) encrypted under the KEK (format 04).
Delimiter	1 A	Value ';'. Optional; if present, the following field must be present.
Padding Mode	1 N	Only present if above Delimiter is present. '0': Append '00' padding if CRT component blocks, or private modulus and exponent blocks, require it to become a multiple of 8 bytes. *Note: This is the default for Exponent/Modulus format if Padding Mode is not present.* '1': For DES KEK, append either 4 byte (8000 0000) or 8 byte mandatory padding (8000 0000 0000 0000) to CRT components or private modulus and exponent so the result is a multiple of 8 bytes. For AES KEK, append either 4 byte (8000 0000) , 8 byte (8000 0000 0000 0000), 12 byte (8000 0000 0000 0000 0000 0000) or 16 byte (8000 0000 0000 0000 0000 0000 0000 0000) mandatory padding to CRT components or private modulus and exponent so the result is a multiple of 16 bytes. *Note: This mode should only be used if the proper key sizes are used.* '2': Append mandatory single byte of '80' followed by '00' bytes, as required, to make CRT component blocks, or private modulus and exponent blocks, a multiple of 8 bytes. *Note: The default for CRT format, if Padding Mode is not present, is a single byte of 80 followed by 00 bytes to make a multiple of 8, ONLY IF, it is not already a multiple of 8.*
KEK		The Key Encryption Key, used to encrypt the Card Private Key components.
	32 H or 'U' + 32 H or 'T' + 48 H	For a Variant LMK, the KEK is encrypted under LMK 24-25/1.
	or 'S' + n B	For a Key Block LMK, the KEK must comply with: **Key Usage**: '54'; **Algorithm**: 'T', 'A'; **Mode of Use**: 'B', 'E', 'D', 'N'
If Card Private Key Output Format = '04' or '05', the following fields must be present as noted below:		
Encrypt Mode	1 N	Mode used to encrypt the Card Private Exponent and Modulus and, if Key Output Format is 05, also the CRT components: '0': ECB mode '1': CBC mode
IV	8 B or 16 B	Initialisation Vector. Only present if Encrypt Mode = '1'. If KEK algorithm = 'A' and Encrypt Mode ='1', the IV will be 16 bytes otherwise 8 bytes.
Delimiter	1 A	Value ';'. Optional; if present, the following field must be present.
Card Private Modulus	1 N	Only present if the previous delimiter was specified.

and Private Exponent Length Bytes Mode		NOTE: This option does not change the format of the encrypted Private Modulus and Private Exponent fields. It only controls whether the encrypted output data is preceded by clear length bytes. If neither this parameter nor the previous delimiter is present, no length bytes will be added in the output. '0': No length bytes are prepended to the encrypted private modulus and private exponent output fields (default). '1': Length bytes are prepended to the encrypted private modulus and private exponent output fields.
Length Bytes	1 N	The number of bytes that are used to specify the length of the key components before component encryption. See Appendix H – Private Key Encodings for format description. Valid entries are '0', '1' or '2'. If this value is zero then no length parameter must be present in the key component block output. For Card Private Key Output Format = '03', a length byte of '1' will automatically be used. (Applies to all key components – modulus, exponent, CRT components.)
If Mode Flag = '0' or '2', the following 3 fields must be present as noted below:		
Card Public Exponent Length	4 N	Optional. Length, in bits, of the Card Public Exponent. Must be supplied if Card Public Exponent is present in command message.
Card Public Exponent	n B	Optional. If supplied, it must follow guidelines described in the command notes. If not supplied then a default exponent of 65537 will be used.
Delimiter	1 A	Mandatory Delimiter. Value ';'.
If Mode Flag = '1', the following 4 fields must be present as noted below:		
Card Private Key Length	4 N	For a Variant LMK, this is the length of the Card Private Key field.
	or 4 H	For a Key Block LMK, this field is ignored and should be set to 'FFFF'.
Card Private Key		The Card Private Key.
	n B	For a Variant LMK, the Card Private Key is encrypted under LMK 34-35.
	or 'S' + n B	For a Key Block LMK, the Card Private Key must comply with: **Key Usage** / **Algorithm** / **Mode of Use** '04' / 'R' / 'S', 'N'
Delimiter	1 A	Mandatory Delimiter. Value ';'
Card Public Key	n B	Card Public Key, DER encoded (unsigned) in ASN.1 format (sequence of modulus and exponent)
The following fields must be present as noted below:		
Hash Identifier	2 N	Identifier of algorithm used to hash data: '01': SHA-1
Signature Identifier	2 N	Identifier of signature algorithm: '01': RSA
PAN	20 H	Application PAN data to be included in certificate. Data supplied is left justified and padded on the right with hex 'F's.
Certificate Expiration Date	4 N	Expiration date of certificate in Month and Year (MMYY) format to be included in the certificate.
Certificate Serial No.	6 H	Certificate tracking number to be included in the certificate.
Data Length	3 N	Length, in bytes, of Static Authentication Data field. Only present if Certificate Type = '0'.
Static Authentication Data	n B	Static authentication data. Only present if Certificate Type = '0'.
Terminator	1 A	Mandatory Delimiter. Value ';'.
Issuer Private Key Flag	2 N	Flag to indicate location of the Issuer's Private Key: If Flag = '99' use Private Key provided with command else Flag = index of stored Private Key.

<table>
<tr><td colspan="3">If Issuer Private Key Flag = '99', the following 2 fields must be present:</td></tr>
<tr><td>Issuer Private Key Length</td><td>4 N</td><td>For a Variant LMK, this is the length of the Issuer Private Key field.</td></tr>
<tr><td></td><td>or
4 H</td><td>For a Key Block LMK, this field is ignored and should be set to 'FFFF'.</td></tr>
<tr><td>Issuer Private Key</td><td></td><td>The Issuer Private Key.</td></tr>
<tr><td></td><td>n B</td><td>For a Variant LMK, the Issuer Private Key is encrypted under LMK 34-35.</td></tr>
<tr><td></td><td>or
'S' + n B</td><td>For a Key Block LMK, the Issuer Private Key must comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'03'</td><td>'R'</td><td>'S', 'N'</td></tr>
</table></td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value '%'. Optional; if present, the following field must be present.</td></tr>
<tr><td>LMK Identifier</td><td>2 N</td><td>LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.</td></tr>
<tr><td>End Message Delimiter</td><td>1 C</td><td>Must be present if a message trailer is present. Value X'19.</td></tr>
<tr><td>Message Trailer</td><td>n A</td><td>Optional. Maximum length 32 characters.</td></tr>
</table>

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'KP'.
Error Code	2 N	'00': No error '04': Invalid Issuer Private Key Flag '05': Invalid Scheme ID, Cert. Type or Mode Flag '06': Invalid Hash or Signature Identifier '09': Invalid Length Bytes value '10': KEK Parity Error '52': Invalid private key output format '53': Invalid Encrypt Mode '54': Invalid Padding Mode '58': Mandatory padding does not result in multiple of 8 bytes '60': Invalid Private Modulus and Exponent Length Bytes Mode 'DA': KEK – key block error 'ED': Issuer Private Key – key block error or a standard error code. **For Mode = '0' or '2' only:** '03': Key Length error '07': Public exponent length error '08': Supplied public exponent value is not allowed '51': Invalid RSA Key Mode Flag **For Mode = '1' only:** '02': Public Key does not conform to encoding rules '55': Card Private Key Length error '56': Card Private Key error '57': Public key / Private key pair mismatch '59': Private Key provided is not Type 3 (ICC) 'E8': Card Private Key – key block error 'E9': Card Public Key – key block error
If Error Code = 'DA', 'E8', E9' or 'ED', the following field will be present:		
Additional Error Code	2 A	The key block specific error code

If the Card Private Key Output format = '03' or '05', the following 6 fields will be present:		
Card Private Key Component Length	1 B	Length, in bytes, of each of the following 5 fields.
p (KEK)	n B	Prime p encrypted under the KEK.
q (KEK)	n B	Prime q encrypted under the KEK.
d1 (KEK)	n B	d1 = d mod (p-1) encrypted under the KEK.
d2 (KEK)	n B	d2 = d mod (q-1) encrypted under the KEK.
q-1 mod p (KEK)	n B	Modular inverse of q encrypted under the KEK.
If the Card Private Key Output format = '04' or '05', the following 2 or 4 fields will be present:		
Card Private Exponent Encrypted Length	2 B	Only present if Card Private Modulus and Exponent Length Bytes Mode = '1'. The length of the entire encrypted Private Exponent Block.
Card Private Key Exponent (KEK)	n B	Card Private Key component formatted in Private Key Exponent/Modulus format (see Appendix H – Private Key Encodings, encrypted under the KEK
Card Private Modulus Encrypted Length	2 B	Only present if Card Private Modulus and Exponent Length Bytes Mode = '1'. The length of the entire encrypted Private Modulus Block.
Card Private Key Modulus (KEK)	n B	Card Private Key modulus formatted in Private Key Exponent/Modulus format (see Appendix H – Private Key Encodings), encrypted under the KEK
The following fields will always be present:		
Card (ICC) Certificate length	2 H	Length, in bytes, of the Card Certificate in the next field.
Card (ICC) Certificate	n B	Signed Card Certificate. Refer to Appendix G - Format of Card (ICC) Public Key Certificate for format.
Card (ICC) Public Key Remainder length	2 H	Length, in bytes, of the Public Key Remainder in the next field. May indicate zero if $N_{IC} <= N_I - 42$.
Card (ICC) Public Key Remainder	n B	Card Public Key Remainder. If the above field indicates zero length (because $N_{IC} <= N_I - 42$), this field will not be present.
Card (ICC) Public Key Exponent Length	2 H	Length, in bytes, of the Public Key Exponent in the next field.
Card (ICC) Public Key Exponent	n B	Card Public Key Exponent.
Card (ICC) Public Key Modulus Length	4 H	Length, in bytes, of the Public Key Modulus in the next field.
Card (ICC) Public Key Modulus	n B	Card Public Key Modulus
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Notations: N_I = Length of the Issuer Public Key Modulus
N_{IC} = Length of the Card (ICC) Public Key Modulus

Import a Certification Authority Self-Signed Certificate

Variant LMK ☑	Key Block LMK ☑
License: **HSM9-LIC002** & **HSM9-LIC016**	
Authorization: **Required** Activity: **import.rsa.host**	

Function: Validate a Certification Authority (CA) public key certificate and return the CA Public Key with its associated MAC and the certificate Expiration Date. For Mastercard, the function will also return the Certificate Serial Number and the Hash for verification of the transferred Public Key.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'KK'.
Scheme ID	1 N	'0': Visa VSDC '1': Mastercard '2': American Express AEIPS V4.1 '3': JCB '4': Union Pay '5': Discover
Certificate Length	4 N	Length, in bytes, of the CA Self-Signed Certificate.
CA Self-Signed Certificate	n B	CA Self-Signed Certificate comprised of the Clear Data and the CA Public Key Certificate. See Appendix L, M, or N depending on Scheme ID.
Delimiter	1 A	Mandatory Delimiter. Value ';'.
Authentication Data	n A	Optional. Additional data to be included in the CA Public Key MAC calculation (must not include ';').
Delimiter	1 A	Only present if the previous field above is present. Delimiter. Value ';'.
Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
If Key Block LMK, the following fields must be present:		
Delimiter	1 A	Value '#'.
Key Version Number	2 N	Key Version Number field, to be included in the key block header; permitted values: '00' to '99';
Exportability	1 A	Exportability field, to be included in the key block header; only permitted values are 'N', 'E' or 'S';
Number of Optional Blocks	2 N	Number of Optional Blocks specified below; permitted values '00' to '08'; must be present if the above Delimiter is present.
For each optional block, the following three fields must be specified. Note: if the Number of Optional Blocks = '00', then none of the following three fields should be present.		
Optional Block Identifier	2 A	Optional Block Identifier; any permitted value except 'PB'.
Optional Block Length	2 H	Number of characters in the optional block (including identifier and length); permitted values X'04 to X'FF; if value = X'04, then the following field is not present.
Optional Block Data	n A	Optional block data.

Field	Length & Type	Details
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'KL'.
Error Code	2 N	'00': No error '05': Invalid Scheme ID '06': Invalid Hash or Signature algorithm indicator '07': Certificate Hash validation failure '08': Mismatch between common clear and signed certificate data '80': Certificate length error '81': Invalid Certificate Format (Header, Trailer or Format value) or a standard error code
MAC	4 B	For a Variant LMK, the MAC on CA Public Key and Authentication Data calculated using LMK 36-37.
CA Public Key		The CA Public Key, protected by the LMK.
	n B	For a Variant LMK, the CA Public Key, DER encoded (unsigned) in ASN.1 format (sequence of modulus and exponent).
	or 'S' + n B	The CA Public Key will conform to the following: **Key Usage**: '02' | **Algorithm**: 'R' | **Mode of Use**: 'V'
For Scheme ID = '0', '1', '2', '3', '4', the following field will be present:		
Certificate Expiration Date	4 N	Certificate Expiration Date (MMYY) recovered from the CA Public Key Certificate.
For Scheme ID = '1' (Mastercard) the following 3 fields will be present (See Appendix M – Self Signed CA Public Key certificate Format (Mastercard)):		
Certificate Serial Number	6 H	Certificate Serial Number recovered from the CA Public Key Certificate.
Verification Hash Length	2 N	Length in hex characters of hash result in next field. This length will depend on the hash algorithm specified in the certificate. For SHA-1 this length will be 40.
Verification Hash Value	n H	Hash for confirmation of CA Public Key transfer. (Includes ID of Certificate, Public Key Index, Modulus and Exponent)
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

EMV Sign Data

Variant LMK ☑	Key Block LMK ☑
License: **HSM9-LIC016**	
Authorization: **Not required**	

Function: Generate a digital signature over a message, as defined in EMV Book 2, using an RSA Private Key. Return the encrypted message.

Notes: The message will not be padded and must be of the same size as the modulus length of provided Private Key or an error will be returned.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'IK'.
Mode Flag	1 N	'0': No formatting or padding of message data
Signature Identifier	2 N	Identifier of signature algorithm: '01': RSA
Message Length	4 N	Length, in bytes, of message data.
Message Data	n B	Message data for signature
Delimiter	1 A	Value ';'. Used to indicate the end of the message data field.
Private Key Flag	2 N	Flag to indicate location of the Private key for signature: 00' ... '20' : index of stored private key '99' : use private key provided with command
If Private Key Flag = '99', the following 2 fields must be present:		
Private Key Length	4 N	For a Variant LMK, this is the length of the Private Key field.
	or 4 H	For a Key Block LMK, this field is ignored and should be set to 'FFFF'.
Private Key		The Private Key.
	n B	For a Variant LMK, the Private Key is encrypted under LMK 34-35.
	or 'S' + n B	For a Key Block LMK, the Private Key must comply with: **Key Usage**: '03' / **Algorithm**: 'R' / **Mode of Use**: 'S', 'N'
Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'IL'.
Error Code	2 A	'00': No error '04': Invalid Private Key Flag '05': Invalid Mode Flag '06': Invalid Signature Identifier '27': Private Key / message length mismatch 'D1': Private Key – key block error or a standard error code
If Error Code = 'D1', the following field will be present:		
Additional Error Code	2 A	The key block specific error code.
If Error Code = '00', the following 2 fields will be present:		
Message Length	4 N	Length, in bytes, of signed message data.
Message Data	n B	Signed message data.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

EMV Recover Data

Variant LMK ☑	Key Block LMK ☑
License: **HSM9-LIC016**	
Authorization: **Not required**	

Function: Recover message data from a digitally signed message, as defined in EMV Book 2, using an RSA Public Key. Return the decrypted message.

Notes: No verification, checking, or removal of padding, will be performed on the decrypted message data.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'IM'.
Mode Flag	1 N	'0': No formatting or padding of message data
Signature Identifier	2 N	Identifier of signature algorithm: '01': RSA
Message Length	4 N	Length, in bytes, of message data.
Message Data	n B	Message data for signature
Delimiter	1 A	Value ';'. Used to indicate the end of the message data field.
Public Key MAC	4 B	Only present if using a Variant LMK. MAC on Public Key and Authentication Data calculated using LMK 36-37.
Public Key		Public Key, DER encoded (unsigned) in ASN.1 format (Sequence of modulus, exponent).
	n B	For a Variant LMK, the Public Key, DER encoded (unsigned) in ASN.1 format (sequence of modulus and exponent).
	or 'S' + n B	For a Key Block LMK, the Public Key must comply with: **Key Usage**: '02' / **Algorithm**: 'R' / **Mode of Use**: 'V', 'N'
Authentication Data	n A	Only present if using a Variant LMK. Optional; additional data to be included in the Public Key MAC calculation (must not include ';').
Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'IN'.
Error Code	2 A	'00': No error '01': MAC verification failure '05': Invalid Mode Flag '06': Invalid Signature Identifier '08': Public Key does not conform to encoding rules 'D2': Public Key – key block error or a standard error code
If Error Code = 'D2', the following field will be present:		
Additional Error Code	2 A	The key block specific error code.
If Error Code = '00', the following 2 fields will be present:		
Recovered Message Length	4 N	Length, in bytes, of the Recovered Message Data.
Recovered Message Data	n B	Recovered message data.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

MULTOS Card Data Preparation Commands

Use of these commands requires the optional MULTOS Card data Preparation license.

The HSM provides the following host commands to support the building of MULTOS Application Load Units (ALUs) and the associated key management:

Import MULTOS Transport Key Certifying Key

Variant LMK ☑	Key Block LMK ☒
License: **HSM9-LIC023**	
Authorization: **Not required**	

Function: To import a MULTOS Transport Key Certifying Key (TKCK) from MULTOS format and return the public key in DER encoded ASN.1 format (TKCK_PK).

Notes: The exponent is always assumed to be 3.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Will be returned to the Host unchanged
Command Code	2 A	Value 'I2'
Mode Flag	1 N	Flag to indicate TKCK format '1': TKCK format (See Appendix J – MULTOS Transport Key Certifying Key File Format) *This field can be expanded in the future to accommodate other certificate types.*
TKCK Length	4 N	Length, in bytes, of the next field
TKCK	n B	MULTOS Transport Key Certifying Key
Delimiter	1 A	Value ';'
Authentication Data	n A	Optional. Additional data to be included in the Public Key MAC calculation (must not include ';')
Delimiter	1 A	Mandatory Delimiter. Value ';'
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Will be returned to the Host unchanged
Response Code	2 A	Value 'I3'
Error Code	2 N	'00': No error '04': Invalid Mode Flag '05': TKCK format, data, or hash error '80': TKCK length error or a standard error code.
MAC TKCK_PK	4 B	MAC on Transport Key Certifying Key and authentication data, calculated using LMK 36-37
TKCK_PK	n B	Transport Key Certifying Key in ASN.1 DER encoded format
TKCK Identifier	4 H	TKCK Identifier
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Import MULTOS Hash Modulus Key

Variant LMK ☑	Key Block LMK ☒
License: **HSM9-LIC023**	
Authorization: **Not required**	

Function: To import a MULTOS Hash Modulus Key from MULTOS format and return the public key in DER encoded ASN.1 format.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Will be returned to the Host unchanged
Command Code	2 A	Value 'I4'
Mode Flag	1 N	Reserved. Must be 0.
Hash Modulus Length	4 N	Length, in bytes, of next field
Hash Modulus	n B	Hash Modulus. See Appendix K for format.
Delimiter	1 A	Value ';'
Public Exponent Length	4 N	Optional. Length, in bits, of the Public Exponent. Must be supplied if Public Exponent is present in command message.
Public Exponent	n B	Optional. If supplied then it must be odd; if not supplied then a default exponent of 3 will be used.
Delimiter	1 A	Mandatory Delimiter. Value ';'
Authentication Data	n A	Optional. Additional data to be included in the MAC calculation over the Hash Modulus (must not contain ';')
Delimiter	1 A	Mandatory Delimiter. Value ';'
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Will be returned to the Host unchanged
Response Code	2 A	Value 'I5'
Error Code	2 N	'00': No error '04': Invalid Mode Flag '05': Hash Modulus format, data, or hash error '07': Public exponent length error '08': Invalid public exponent '80': Hash Modulus Length error or a standard error code.
MAC	4 B	MAC on Hash Modulus key and authentication data, if supplied, calculated using LMK 36-37
Hash Modulus	n B	Hash Modulus Key in ASN.1 DER encoded format
Hash Modulus Identifier	4 H	Hash Modulus Identifier
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Will only be present if in the command message. Maximum length 32 characters

Translate MULTOS KTU

Variant LMK ☑	Key Block LMK ☒
License: **HSM9-LIC023** & **HSM9-LIC002**	
Authorization: **Not required**	

Function: To translate a Key Transformation Unit (KTU Prime) from encryption under a Key Encryption Key (KEK) to the standard MULTOS KTU format encrypted under either an RSA public key or diversified key (Step One).

Notes:

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Will be returned to the Host unchanged
Command Code	2 A	Value 'I6'
Version Flag	1 N	Flag to indicate MULTOS version: '0': MULTOS v3.0 '1': MULTOS v4.0 '2': MULTOS v4.0 with hash verification '3': MULTOS Step One
KEK	'U' + 32 H or 'T' + 48 H	KEK encrypted under LMK 24 25/1.
KTU Length	3 N	Length, in bytes, of next field (maximum of 256)
KTU Prime	n B	KTU encrypted under the KEK
Delimiter	1 A	Value ';'
If Version Flag = 0, 1 or 2, the following 7+4 fields must be present:		
Tkck_pk MAC	4 B	MAC on Transport Key Certifying Key, and optional authentication data, calculated using LMK 36-37
tkck_pk	n B	Transport Key Certifying Key in ASN.1 DER encoded format.
TKCK_PK Authentication Data	n A	Optional. Additional data included in the MAC calculation (must not include ';')
Delimiter	1 A	Mandatory Delimiter. Value ';'
MCD_PK_C Length	3 N	Length, in bytes, of the next field
MCD_PK_C	n B	Card Public Key Certificate (see Appendix I)
Delimiter	1 A	Value ';'
If Version Flag = 2, the following 4 fields must be present:		
Hash Modulus MAC	4 B	MAC on Hash Modulus, and optional authentication data, calculated using LMK 36-37
Hash Modulus	n B	Hash Modulus Key, in ASN.1 DER encoded format
Hash Modulus Authentication Data	n A	Optional; additional data included in the MAC calculation (must not include ';')
Delimiter	1 A	Mandatory Delimiter. Value ';'
If Version Flag = 3, the following field must be present:		
MK-KE	'U' + 32 H or 'T' + 48 H	The Master KTU Encipherment key encrypted under LMK 24-25/8.
Enablement_Data_ Production_Date	1 B	Date on which Enablement Data for the target card was created. The byte corresponds to the number of months since January 1998.
MCD_NO	8 B	MCD Number for the KTU
MCD_ID	6 B	MCD ID used to generate diversified key.

LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Will be returned to the Host unchanged
Response Code	2 A	Value 'I7'
Error Code	2 N	'00': No error '01': TKCK MAC verification failure '04': Invalid ALU Identifier '07': TKCK_PK does not conform to encoding rules '08': Invalid TKCK_PK length '09': KTU Prime length error '10': KEK parity error '11': MK-KE parity error '50': Hash Modulus does not conform to encoding rules '51': Hash Modulus MAC verification failure '52': Card Certificate Hash validation failure '53': Card Public Key modulus length mismatch '54': Invalid KTU header '55': KTU too long for Card Public Key '56': TKCK_PK – MCD_PK ≥ 56 '80': MCD_PK_C length error or a standard error code.
KTU Length	3 N	Length in bytes of the next field
KTU	n B	Standard MULTOS KTU, encrypted under the smart card RSA public key
End Message Delimiter	1 C	Will only be present if present in the command message. Value X'19
Message Trailer	n A	Will only be present if in the command message. Maximum length 32 characters

MULTOS ALU Generator - Introduction

The MULTOS ALU Generator function consists of a series of sub commands (described in the following sections) in the I8 command which are used in the appropriate sequence to generate unprotected, protected, or confidential MULTOS Application Load Units (ALUs).
Use of the I8 command set requires license HSM9-LIC023. Those sub commands which use the RSA algorithm also require license HSM9-LIC002.

Typical sequencing of sub-commands to build an ALU for an EMV application would be as follows:

Sub Command	Description
Allocate ALU Area	Create an initialized ALU Area
Load Block (Code)	Load application code into ALU Code Block
Load Block (FCI)	Load FCI data into ALU FCI Block
Load Block (DIR)	Load DIR data into ALU DIR Block
Load Block (Data)	Load static application data in ALU Data Block
For each card:	
Load Clear Data (TLV)	*Load personalized data items into ALU Data Block*
Load Cipher Data (PIN)	*Load enciphered PIN into ALU*
Load Cipher Data (DES Key)	*Load enciphered EMV UDKs into ALU*
Load Cipher Data (RSA key)	*Load EMV ICC RSA private keys into ALU*
Generate Checksum ()	*Generate checksums over data areas*
Encrypt Area ()	*Encipher areas under KTU keys*
Generate Signature ()	*Sign or MAC the ALU (protected ALU)*
Generate KTU ()	*Encipher KTU under card key (confidential ALU)*
Return ALU ()	*Return completed ALU to host*
Release ALU Area ()	Delete ALU and release work areas

The generic structure of the I8 command is described in the following table. The parameters, error codes, and returned data peculiar to each sub command are described in the subsequent sections on each sub-command.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'I8'
Sub Command Code	2 N	'01': Allocate ALU Area '02': Load Block '03': Load Clear Data '04': Load Cipher Data '05': Not used. Reserved '06': Generate Checksum '07': Encrypt Area '08': Generate Signature '09': Generate KTU '10': Return ALU '11': Release ALU Area
Sub Command parameters are defined in the sections below.		
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'I9'
Error Code	2 N	'00': No error '02': Invalid sub command value or a specific subcommand error code or a standard error code.
Refer to sections below for error codes and parameters returned by each Sub Command Code		
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

MULTOS ALU Generator – Allocate ALU Area

Variant LMK ☑	Key Block LMK ☒
License: **HSM9-LIC023**	
Authorization: **Not required**	

Function: Initial command to allocate and lock a memory block for building the Application Load Unit. All subsequent subcommands reference this block using an ALU identifier. The memory block is released using the Release ALU Area sub command. Multiple ALU areas can be assigned and individually referenced using the identifier.

Notes: If the ALU Identifier has already been used or there is not enough memory available to assign to an ALU area, an error will be returned.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'I8'
Sub Command Code	2 N	'01': Allocate ALU Area
ALU Identifier	8 A	Unique identifier for allocated ALU area
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'I9'
Error Code	2 N	'00': No error '02': Invalid sub command value '04': Invalid ALU Identifier (identifier already in use) '06': Memory allocation error or a standard error code.
Assigned ALU Areas	2 N	Total number of assigned ALU areas, including this one. Note: This parameter is returned even if Error Code 06 is returned, except it will only contain the number of ALU areas already allocated.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

MULTOS ALU Generator – Load Block

Variant LMK ☑	Key Block LMK ☒
License: **HSM9-LIC023**	
Authorization: **Not required**	

Function: This sub-command is used to load initialised or static blocks into the ALU prior to personalisation, for example Code, FCI and DIR blocks. A default Data Block may also be loaded which can then be personalised for each card using the Load Clear Data and Load Cipher Data sub commands.

Notes: The Data Block loaded by this function will be saved and used to re-initialize the Data Block area when the completed ALU is returned so that multiple ALUs can be built using a common data template. See Release ALU sub-command for additional details. The loading of a Data Block will clear all activities on a previously loaded Data Block.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'I8'
Sub Command Code	2 N	'02': Load Block
ALU Identifier	8 A	Unique identifier for allocated ALU area
Block Type	1 N	Type of block being loaded '1': DIR '2': FCI '3': Code '4': Data
Block Length	4 H	Length of Block Data
Block Data	n B	Block Data to load into ALU
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'I9'
Error Code	2 N	'00': No error '02': Invalid sub command value '03': Illegal operation, specified ALU is in use '04': Invalid ALU Identifier (identifier not known) '05': Invalid Block Type '06': Memory allocation error or a standard error code.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

MULTOS ALU Generator – Load Clear Data

Variant LMK ☑	Key Block LMK ☒
License: **HSM9-LIC023**	
Authorization: **Not required**	

Function: This sub command is used to personalise data elements in the Data, FCI or DIR blocks. The data elements personalised with this function are only clear data elements. Enciphered data elements, such as PINs and Keys, are loaded using the Load Cipher Data sub command.

Notes: If the Block Type is = 4 and the clear data is being loaded over any part of cipher data which has already been loaded, or the Encrypt Area sub command has been executed over any part of the Data Block, this sub command will return an error.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'I8'
Sub Command Code	2 N	'03': Load Clear Data
ALU Identifier	8 A	Unique identifier for allocated ALU area
Block Type	1 N	Type of block being loaded '1': DIR '2': FCI '4': Data
Offset	4 H	Indicates the offset from the start of the Block (Note: An offset of 0 indicates the first byte in the Block)
Data Length	4 H	Length of next field
Data	n B	Data to load at Offset
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'I9'
Error Code	2 N	'00': No error '02': Invalid sub command value '03': Illegal operation, specified ALU is in use '04': Invalid ALU Identifier (identifier not known) '05': Invalid Block Type '06': Memory allocation error '07': Data range error, exceeds maximum block length '09': Illegal operation, clear data overlaps loaded cipher data or encryption of Data Block has already started or a standard error code.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

MULTOS ALU Generator – Load Cipher Data

Variant LMK ☑	Key Block LMK ☒
License: **HSM9-LIC023**	
Authorization: **Not required**	

Function: This sub command is used to personalise enciphered data elements in the Data block, for example PINs and Keys. Clear data elements are loaded using the Load Clear Data sub command.

Input for the different types of enciphered data elements in this command needs to be in the following formats:

PIN blocks encrypted under ZPK (formats will not be checked)

Use Translate PIN commands JG (LMK to ZPK) or CC (ZPK to ZPK)

Triple DES keys encrypted under a KEK

Use Derive Card Unique DES Keys command (KI)

RSA keys as 5 CRT components encrypted under a KEK

Use Generate Card RSA Key Set and Public Key Certificate command (KO)

Data encrypted under a DEK using ECB or CBC format

Use Encrypt Data Block command 'M0'

Notes: For CRT components, the function will remove any length and padding bytes from the components and puts them in the specified location as defined by the Component Placement Flag.

If the Encrypt Area sub command has been executed over any part of the Data Block, this sub command will return an error.

Compatibility Note: In this revision, parameters for Cipher Data Type = '3' have been changed and it will not be backwards compatible. Previously, there was no "Length Bytes" parameter in the command message and therefore it was not possible to remove it, or the padding, from the CRT components. This has now been addressed.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'I8'
Sub Command Code	2 N	'04': Load Cipher Data
ALU Identifier	8 A	Unique identifier for the ALU to receive the data.
Block Type	1 N	Reserved, Must be 4.
Cipher Data Type	1 N	'1': PIN Block encrypted under ZPK '2': Triple DES Key encrypted under KEK '3': RSA private key in CRT format encrypted using CBC under KEK (See Appendix H – Private Key Encoding) '4': Data ECB or CBC encrypted under DEK '5': RSA private key in CRT format encrypted using ECB under KEK (See Appendix H – Private Key Encoding)
Encryption Key	'U' + 32 H or 'T' + 48 H	Encryption Key encrypted under the appropriate LMK. KEK encrypted under LMK 24-25/1. ZPK encrypted under LMK 06-07. DEK encrypted under LMK 32-33.

Field	Length & Type	Details
Offset	4 H	Indicates the offset from the start of the Data Block (Note: An offset of 0 indicates the first byte in the block)
If Cipher Data Type = '1', the following field must be present:		
PIN Block	16 H	PIN block encrypted under the ZPK
If Cipher Data Type = '2', the following 2 fields must be present:		
DK (KEK)	'X' + 16 B	The derived unique key encrypted under the KEK
Atalla Variant	1 / 2 N	Optional. Only required if DK was encrypted under the KEK with Atalla variant
If Cipher Data Type = '3' or '5', the following 8 fields must be present:		
Private Key Component Length	1 B	Length, in bytes, of each of the following 5 fields.
p (KEK)	n B	Prime p encrypted under KEK using triple DES.
q (KEK)	n B	Prime q encrypted under KEK using triple DES.
d1 (KEK)	n B	d1 = d mod (p-1) encrypted under KEK using triple DES.
d2 (KEK)	n B	d2 = d mod (q-1) encrypted under KEK using triple DES.
q-1 mod p (KEK)	n B	Modular inverse of q encrypted under KEK using triple DES.
Length Bytes	1 N	Number of bytes used to specify the length of the key component in the field. See Appendix J for format description. Valid entries are 0, 1 or 2. If this value is zero the component will be expected to have mandatory padding consisting of a single byte of '80' followed by '00' bytes, as required, to make the CRT component block a multiple of 8 bytes.
Component Placement Flag	1 N	'0': Decrypted components will be concatenated in the following order - d1, d2, p, q, q-1 mod p. '1': Decrypted components will be concatenated in the following order - d1, d2, p, q, q-1 mod p - with p, q and q-1 mod p shifted as a block, if necessary, to begin on an 8 byte boundary with binary 0s filling any vacated space
If Cipher Data Type = '4', the following 4 fields must be present:		
Encryption Mode	1 N	'0': ECB '1': CBC
IV	16 H	Only present if Encryption Mode = 1 IV value
Message Length	4 H	Length of the following field in bytes
Encrypted Message	n B	Data encrypted under the DEK
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'I9'
Error Code	2 N	'00': No error '02': Invalid sub command value '03': Illegal operation, specified ALU is in use '04': Invalid ALU Identifier (identifier not known) '05': Invalid Block Type or Cipher Data Type '06': Memory allocation error '07': Data range error, exceeds maximum block length '08': Invalid Component Placement Flag or Encryption Mode '09': Illegal operation, encryption of Data Block has already started '10': Encryption Key Parity Error '11': DK Parity Error '50': Invalid CRT component length byte contents '51': Invalid CRT component padding characters '80': CRT components, or Encrypted Message, length error or a standard error code.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

MULTOS ALU Generator – Generate Checksum

Variant LMK ☑	Key Block LMK ☒
License: **HSM9-LIC023**	
Authorization: **Not required**	

Function: This sub command calculates a MULTOS checksum over sections of the Data Block. Checksums may be computed over plain text areas and areas that contain deciphered elements such as PINs and Keys. This command is usually called once the Data section has been personalised.

Notes: Checksums generated over data loaded using the Load Cipher Data sub command will require encryption if the 'Protect MULTOS Cipher Data Checksums' HSM security parameter is set to 'Yes'. If encryption is not required (parameter set to 'No'), checksum calculations must encompass at least the entire data type that was loaded (PIN, Key, etc.) with the Load Cipher Data sub command. An error will be returned if a checksum is to be generated over partial PINs, Keys, etc.

Unprotected checksums generated over sensitive data such as PIN, Keys, etc. can present a security risk. It is strongly recommended that such checksums be encrypted.

If the Encrypt Area sub command has been executed over any part of the Data Block, this sub command will return an error.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'I8'
Sub Command Code	2 N	'06': Generate Checksum
ALU Identifier	8 A	Unique identifier for the ALU to carry out the operation.
Block Type	1 N	Reserved, Must be 4.
Offset	4 H	Indicates the offset from the start of the Data Block to the start the checksum operation. (Note: An offset of 0 indicates the first byte in the block)
Length	4 H	Number of bytes over which to calculate the checksum.
Checksum Method	1 N	'1': Standard MULTOS checksum.
Checksum IV	8 H	Initial Value for checksum process. Should be set to 0x5AA55AA5 as defined by MULTOS.
Checksum Result Offset	4 H	Indicates offset from the start of the Data Block for placing the calculated 4 byte checksum value (Note: An offset of 0 indicates the first byte in the block)
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'I9'
Error Code	2 N	'00': No error '02': Invalid sub command value '03': Illegal operation, specified ALU is in use '04': Invalid ALU Identifier (identifier not known) '05': Invalid Block Type or Checksum Method '07': Data range error, exceeds block length '08': Checksum result range error, exceeds block length '09': Illegal operation, checksum over partial cipher data, or encryption of Data Block has already started or a standard error code.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

MULTOS ALU Generator – Encrypt Area

Variant LMK ☑	Key Block LMK ☒
License: **HSM9-LIC023**	
Authorization: **Not required**	

Function: This sub command is used to encrypt areas of the Data Block that contain cryptographic data elements such as PINs and Keys that have been loaded using the Load Cipher Data sub command or checksums created using the Generate Checksum sub command if protected checksums are required. This command also creates Area Descriptors that define the encrypted areas for inclusion in the KTU.

Notes: The use of Encryption Methods '01' and '03' require that the 'Enable Single-DES' HSM security parameter be set to Yes or the sub command will return an error.

All data should be loaded and checksums generated before executing the Encrypt Area sub command. Once this command is executed on any area of the Data Block, the loading of clear or cipher data (sub commands 4 and 5) or the generation of checksums (sub command 6) will be prohibited.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'I8'
Sub Command Code	2 N	'07': Encrypt Area
ALU Identifier	8 A	Unique identifier for the ALU to carry out the operation.
Block Type	1 N	Reserved, Must be 4.
Offset	4 H	Indicates the offset from the start of the Data Block to start the encryption operation. (Note: An offset of 0 indicates the first byte in the block)
Length	4 H	Number of bytes to encrypt. Minimum length is 8 bytes. Must be a multiple of 8 bytes.
Encryption Method	2 N	'01': DES-1 CBC with single length DES key (using decrypt operation) '02': Triple DES-1 CBC with double length DES key (using decrypt, decrypt, decrypt operations) The following settings should only be used for StepOne. '03': DES CBC with single length DES key '04': Triple DES CBC with double length DES key
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'I9'
Error Code	2 N	'00': No error '02': Invalid sub command value '03': Illegal operation, specified ALU is in use '04': Invalid ALU Identifier (identifier not known) '05': Invalid Block Type '07': Data range error, exceeds block length '08': Invalid Encryption Method '80': Invalid Length or a standard error code.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

MULTOS ALU Generator – Generate Signature

Variant LMK ☑	Key Block LMK ☒
License: **HSM9-LIC002** & **HSM9-LIC023**	
Authorization: **Not required**	

Function: This sub-command is required for protected ALUs and generates a signature over the Application Unit using either the Application Provider Private Key or the Application Signature Key (for Step One). The signature is generated after sensitive data elements in the Data Block have been encrypted using the Encrypt Area sub command.

Notes: If there are any unprotected data elements remaining in the Data Block (loaded with Load Cipher Data sub command or generated with the Generate Checksum sub command, when protected checksums are required, but not subsequently protected using the Encrypt Area sub command), this command will not generate a signature and will return an error.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'I8'
Sub Command Code	2 N	'08': Generate Signature
ALU Identifier	8 A	Unique identifier for the ALU to carry out the operation.
Version Flag	1 N	Flag to indicate MULTOS version: '0': MULTOS v3.0 '1': MULTOS v4.0 '3': MULTOS Step One
If Version Flag = '0' or '1', the following 8 fields must be present:		
Private Key Flag	2 N	Flag to indicate location of the Private Key; '00' ... '98': index of stored private key '99': use private key provided with command.
Private Key Length	4 N	Optional. Must be present if Private key flag = '99' Length in bytes of the following field
Application Provider Private Key	n B	Optional. Must be present if Private key flag = '99' Private key encrypted under LMK 34-35
Hash Algorithm	1 N	'1': MULTOS Asymmetric Hash This field can be expanded in the future to accommodate other Hash types.
If Hash Algorithm = '1', the following 4 fields must be present:		
Hash Modulus MAC	4 B	MAC on Hash Modulus key and optional authentication data calculated using LMK 36-37.
Hash Modulus	n B	Hash Modulus Key in ASN.1 DER encoded format.
Hash Modulus Authentication Data	n A	Optional; additional data included in MAC calculation (must not include ';')
Delimiter	1 A	Mandatory Delimiter. Value ';'
If Version Flag = '3', the following 3 fields must be present:		
MK-AS	'U' + 32 H or 'T' + 48 H	The Master Application Signature (MAC) key encrypted under LMK 24-25/9.
Key Diversification Flag	1 N	'0': MAC generated using MK-AS '1': MAC generated using diversified key
MCD_ID	6 B	Only present if Key Diversification Flag = 1 MCD ID used to generate diversified key.

Field	Length & Type	Details
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'I9'
Error Code	2 N	'00': No error '02': Invalid sub command value '03': Illegal operation, specified ALU is in use '04': Invalid ALU Identifier (identifier not known) '05': Invalid Version Flag '06': Memory allocation error '07': Invalid Hash Algorithm or Key Diversification Flag '08': Invalid Private Key Flag '09': Illegal signature over unprotected sensitive data '11': MK-AS parity error '50': Hash Modulus does not conform to encoding rules '51': Hash Modulus MAC verification failure or a standard error code.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

MULTOS ALU Generator – Generate KTU

Variant LMK ☑	Key Block LMK ☒
License: **HSM9-LIC002** & **HSM9-LIC023**	
Authorization: **Not required**	

Function: This sub-command is used to generate the Key Transformation Unit for confidential ALUs. The command can generate KTUs encrypted under the specific Card Public key or a KTU Prime encrypted under a temporary key if the target card is unknown. The Translate KTU command may be used to translate from KTU Prime to a card specific KTU once the target card is known.

Notes: If there are any unprotected data elements remaining in the Data Block (loaded with Load Cipher Data sub command or generated with the Generate Checksum sub command, when protected checksums are required, but not subsequently protected using the Encrypt Area sub command), this command will not generate a KTU and will return an error.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'I8'
Sub Command Code	2 N	'09': Generate KTU
ALU Identifier	8 A	Unique identifier for the ALU to carry out the operation.
KTU Type	1 N	'1': KTU Prime, encryption under KEK '2': KTU, encryption under the card public key '3': KTU, encryption under a diversified key (Step One).
Version Flag	1 N	Flag to indicate MULTOS version: '0': MULTOS v3.0 '1': MULTOS v4.0 '2': MULTOS v4.0 with hash verification
Application ID	17 B	Application Identifier consisting of the AID length (1 byte) followed by the AID (n bytes) and padding (0xFF bytes to fill 17 bytes).
If KTU Type = '1', the following field must be present:		
KEK	'U' + 32 H or 'T' + 48 H	KEK encrypted under LMK 24 25/1.
If KTU Type = '2', the following 12 fields must be present:		
Tkck_pk MAC	4 B	MAC on Transport Key Certifying Key, and optional authentication data, calculated using LMK 36-37.
tkck_pk	n B	Transport Key Certifying Key in ASN.1 DER encoded format.
TKCK_PK Authentication Data	n A	Optional. Additional data included in the MAC calculation (must not include ';')
Delimiter	1 A	Mandatory Delimiter. Value ';'
MCD_PK_C Length	3 N	Length, in bytes, of the next field
MCD_PK_C	n B	MULTOS Card Public Key Certificate (See Appendix I – MULTOS Card Public Key Certificate Format
Delimiter	1 A	Value ';'
If Version Flag = '2', the following 4 fields must be present:		
Hash Modulus MAC	4 B	MAC on Hash Modulus, and optional authentication data, calculated using LMK 36-37.
Hash Modulus	n B	Hash Modulus Key, in ASN.1 DER encoded format

Field	Length & Type	Details
Hash Modulus Authentication Data	n A	Optional; additional data included in the MAC calculation (must not include ';')
Delimiter	1 A	Mandatory Delimiter. Value ';'
If KTU Type = 3, the following 3 fields must be present:		
MK-KE	'U' + 32 H or 'T' + 48 H	The Master KTU Encipherment key encrypted under LMK 24-25/8.
Enablement_Data_ Production_Date	1 B	Date on which Enablement Data for the target card was created. The byte corresponds to the number of months since January 1998.
MCD_NO	8 B	MCD Number for the KTU
MCD_ID	6 B	MCD ID used to generate diversified key.
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'I9'
Error Code	2 N	'00': No error '01': TKCK MAC verification failure '02': Invalid sub command value '03': Illegal operation, specified ALU is in use '04': Invalid ALU Identifier (identifier not known) '05': Invalid KTU Type or Version Flag '06': Memory allocation error '07': TKCK_PK does not conform to encoding rules '08': No Area Descriptors defined '09': Unprotected sensitive data remains in Data Block '10': KEK parity error '11': MK-KE parity error '50': Hash Modulus does not conform to encoding rules '51': Hash Modulus MAC verification failure '52': Card Certificate Hash validation failure '53': Invalid TKCK_PK length '54': Card Public Key modulus length mismatch '55': KTU too long for Card Public Key '56': TKCK_PK – MCD_PK ≥ 56 '80': MCD_PK_C length error or a standard error code.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

MULTOS ALU Generator – Return ALU

Variant LMK ☑	Key Block LMK ☒
License: **HSM9-LIC023**	
Authorization: **Not required**	

Function: This sub command returns the completed ALU to the Host.

It will then delete the internal Signature and KTU areas, if present, and re-initialize the Data Block area to the last one loaded using the Load Block sub command in case that a new ALU based on the same common data needs to be created.

Notes: If there are any unprotected data elements remaining in the Data Block (loaded with Load Cipher Data sub command or generated with the Generate Checksum sub command, when protected checksums are required, but not subsequently protected using the Encrypt Area sub command), or if Area Descriptors have been generated with the Encrypt Data sub command and a KTU has not been created, this command will return an error and no further processing with take place.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'I8'
Sub Command Code	2 N	'10': Return ALU
ALU Identifier	8 A	Unique identifier for the ALU to be returned.
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'I9'
Error Code	2 N	'00': No error '02': Invalid sub command value '03': Illegal operation, specified ALU is in use '04': Invalid ALU Identifier (identifier not known) '08': KTU has not been created and protected '09': Unprotected sensitive data remains in ALU or a standard error code.
ALU Length	4 H	Length of the ALU
ALU	n B	The completed Application Load Unit
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

MULTOS ALU Generator – Release ALU

Variant LMK ☑	Key Block LMK ☒
License: **HSM9-LIC023**	
Authorization: **Not required**	

Function: This sub command releases the allocated memory and frees all allocated resources created during the ALU Generator command.

Notes: It is the responsibility of the calling application to de-allocate all ALU areas once the completed ALU(s) has been successfully retrieved.

A method is provided to clear all ALU areas in case the application has lost track of ALU Identifiers and cannot individually clear them. If the ALU Identifier "ClearAll" (case insensitive) is used in this sub-command, all ALU areas will be released **except** for any ALU areas currently being worked on by another sub-command. This case will only occur in multi-threaded applications where other threads are actively working on ALUs. It is recommended that the "ClearAll" method is only used when there is no work currently taking place on ALUs in order to prevent unexpected release of ALUs.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'I8'
Sub Command Code	2 N	'11': Release ALU
ALU Identifier	8 A	Unique identifier for the ALU to release.
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'I9'
Error Code	2 N	'00': No error '02': Invalid sub command value '03': Illegal operation, specified ALU is in use '04': Invalid ALU Identifier (identifier not known) or a standard error code.
Assigned ALU Areas	2 N	Total number of remaining assigned ALU areas.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Chip Card Personalization Commands

Use of these commands requires the optional General Purpose Card Personalization license.

The HSM provides the following host commands to support the secure operations required for general purpose card personalization:

Establish Secure Session with Chip Card

Variant LMK ☑	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Not required**	

Function: Support the establishment of a secure session with a chip card, using mutual authentication, based on the EMV Common Personalization Specification (CPS) and Global Platform (GP).

The following protocols are supported:

- Secure Channel Protocol 02 (SCP02) "i" = 0x15
- Secure Channel Protocol 02 (SCP02) "i" = 0x55
- Support for the specific process for Mastercard PayPass Magnetic Stripe Cards
- Support for static personalization method where the card application is personalized with a single Personalisation Secret Key (PSK).
- Secure Channel Protocol 03 (SCP03)

Notes: There are two methods under EMV CPS and GP SCP02 that can be used to establish a secure personalization channel with the card application as described below.

Indirect (Explicit) Method:

In this method, the Data Preparation Process does not need to have knowledge of the process used to establish a secure personalization session with a chip card. Therefore, two security zones exist in this process - a security zone between the Data Preparation Process and the Personalization Process and another security zone between the Personalization Process and the chip card.

The Personalization Process establishes a secure session with the chip card by issuing the INITIALISE UPDATE and EXTERNAL AUTHENTICATE commands. Through this process, the session keys and cryptograms required to establish a secure session with a chip card are generated.

Direct (Implicit) Method:

This method is not supported at this time.

For the above methods, this function uses the provided INITIALISE UPDATE responses to generate the data for the EXTERNAL AUTHENTICATE command, as well as derive the keys needed for personalization of the chip card. The keys are returned encrypted under the LMK for use by the Prepare Secure Message for Chip Card command. The Global Platform specific options for Security Level which include the use of an R-MAC are also supported. If an option which includes an R-MAC is defined, the R-MAC key will also be

derived and returned encrypted under the LMK. This key will be handled as a 'ZAK' so that messages from the chip card can be validated using the "Verify MAC" command in the HSM.

For the Mastercard PayPass Magnetic Stripe method, the function will only derive the KD Personalization Key which will be used for authenticating messages to the card and securing sensitive data. The key is returned encrypted under the LMK for use by the Prepare Secure Message for Chip Card command.

C-MAC is a MAC generated by the Host over the APDU command message.

R-MAC is a MAC generated by the Card over the APDU response message.

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">COMMAND MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Subsequently returned to the Host unchanged.</td></tr>
<tr><td>Command Code</td><td>2 A</td><td>Value 'IC'.</td></tr>
<tr><td>Secure Channel Method</td><td>1 N</td><td>'0': Indirect (Explicit) initiation of secure channel by the personalization system, with Card Keys derived from KMC. (Global Platform SCP02 "i" = 0x15 method).
'1': Indirect (Explicit) initiation of secure channel by the personalization system, with Card Keys provided in the command. (Global Platform SCP02 "i" = 0x15 method).
'2': Mastercard PayPass Magnetic Stripe Cards. Perso Key derived from KMC.
'4': Indirect (Explicit) initiation of secure channel by the personalization system, with Card Keys derived from KMC. (Global Platform SCP02 "i" = 0x55 method).
'5': Indirect (Explicit) initiation of secure channel by the personalization system, with Card Keys provided in command. (Global Platform SCP02 "i" = 0x55 method).
'6': Indirect (Explicit) initiation of Open Platform secure channel by the personalization system using a single personalization key PSK. (Static authentication method).
'7': Secure Channel Protocol 03 (SCP03)</td></tr>
<tr><td colspan="3">If Secure Channel Method = '6' or '7', the following field will be present:</td></tr>
<tr><td>Card Key Generation Mode</td><td>1 N</td><td>'0': Card key(s) derived from master KMC key
(not valid for Secure Channel Method '7')
'1': Card keys provided in command</td></tr>
<tr><td colspan="3">If Secure Channel Method = '0', '2' or '4', or if Secure Channel Method = '6' and Card Key Generation Mode = '0', the following 2 fields must be present:</td></tr>
<tr><td rowspan="3">KMC</td><td></td><td>The Master Personalization Key.</td></tr>
<tr><td>32 H
or
'U' + 32 H
or
'T' + 48 H</td><td>For a Variant LMK, the KMC is encrypted under LMK 24-25/2.</td></tr>
<tr><td>or
'S' + n A</td><td>For a Key Block LMK, the KMC must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'E7'</td><td>'T'</td><td>'N'</td></tr>
</table></td></tr>
<tr><td rowspan="3">Derivation Data</td><td></td><td>Data used to generate the card static diversified keys.</td></tr>
<tr><td>6 B</td><td>If Secure Channel Method is '0' or '4', these are the 6 least significant bytes of KEYDATA typically from the response to an INITIALISE UPDATE command to the card. If Secure Channel Method is 2, this field is ignored and set to 6 bytes of 0x00.</td></tr>
<tr><td>16 B</td><td>If Secure Channel Method is '6', this will be the 16 byte PSK</td></tr>
</table>

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td></td><td></td><td>derivation data.</td></tr>
<tr><td colspan="3">If Secure Channel Method = '1' or '5', the following 3 fields must be present:</td></tr>
<tr><td>CK-ENC</td><td></td><td>Card Key for encryption session key generation.</td></tr>
<tr><td></td><td>'U' + 32 H
or
'T' + 48 H</td><td>For Variant LMK, the CK-ENC encrypted under LMK 36-37/3.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a Key Block LMK, the CK-ENC must comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'37'</td><td>'T'</td><td>'X', 'N'</td></tr></table></td></tr>
<tr><td>CK-MAC</td><td></td><td>Card Key for integrity session key generation.</td></tr>
<tr><td></td><td>'U' + 32 H
or
'T' + 48 H</td><td>For Variant LMK, the CK-MAC is encrypted under LMK 36-37/4.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a Key Block LMK, the CK-MAC must comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'38'</td><td>'T'</td><td>'X', 'N'</td></tr></table></td></tr>
<tr><td>CK-DEK</td><td></td><td>Card Key for card data encryption session key generation.</td></tr>
<tr><td></td><td>'U' + 32 H
or
'T' + 48 H</td><td>For Variant LMK, the CK-DEK is encrypted under LMK 36-37/5.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a Key Block LMK, the CK-DEK must comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'39'</td><td>'T'</td><td>'X', 'N'</td></tr></table></td></tr>
<tr><td colspan="3">If Card Key Generation Method = '1', the following 3 fields must be present:</td></tr>
<tr><td>PSK</td><td></td><td>Only present if Secure Channel Method = '6'.
The Personalisation System Key.</td></tr>
<tr><td></td><td>'U' + 32 H</td><td>For a Variant LMK, the PSK is encrypted under LMK 24-25/5.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a Key Block LMK, the PSK must comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'40'</td><td>'T'</td><td>'N'</td></tr></table></td></tr>
<tr><td>CK-ENC</td><td>'S' + n A</td><td>Only present if Secure Channel Method = '7'.
Card Key for cryptograms.
The CK-ENC must comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'37'</td><td>'A'</td><td>'X'</td></tr></table></td></tr>
<tr><td>CK-MAC</td><td>'S' + n A</td><td>Only present if Secure Channel Method = '7'.
Card Key for authentication.
The CK-MAC must comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'38'</td><td>'A'</td><td>'X'</td></tr></table></td></tr>
<tr><td colspan="3">For all values of Secure Channel Method, the following field must be present:</td></tr>
<tr><td>Key Scheme (LMK)</td><td>1 A</td><td>For Variant LMK, Indicates the scheme for encrypting derived keys under the LMK. See the Key Scheme Table in Appendix A of the payShield 9000 Host Programmer's Manual for a list of schemes.
Valid value is 'U'.
For Key Block LMK, this field must be set to 'U'.</td></tr>
<tr><td colspan="3">If Secure Channel Method = '0', '1', '4', '5' or '6', the following 2 fields must be present (SCP02):</td></tr>
<tr><td>Host Challenge</td><td>8 B</td><td>Random number provided by the Personalization System on INITIALISE UPDATE command to the card</td></tr>
</table>

Field	Length & Type	Details
Sequence Counter	2 B	Sequence counter returned by the INITIALISE UPDATE command to the card which is used for derivation of the card session keys
If Secure Channel Method = '7', the following 3 fields must be present (SCP03):		
Card Challenge Mode	1 N	'0': Random. Card challenge and cryptogram passed to command '1': Predictive. Card challenge and cryptogram generated by command
Host Challenge	8 B	Random number provided by Personalization System
Sequence Counter	3 B	Sequence counter returned by the INITIALISE UPDATE command to the card which is used for derivation of the card session keys.
If Secure Channel Method = '0', '1', or '6', the following 2 fields must be present (SCP02 i='15'):		
Card Challenge	6 B	Random number generated by the card in response to the INITIALISE UPDATE command
Card Cryptogram	8 B	Cryptogram generated by the card in response to INITIALISE UPDATE command and used by the Host to authenticate the card
If Secure Channel Method = '7' and Card Challenge Mode = '0', the following 2 fields must be present (SCP03):		
Card Challenge	8 B	Random number generated by the card in response to the INITIALISE UPDATE command
Card Cryptogram	8 B	Cryptogram generated by the card in response to INITIALISE UPDATE command and used by the Host to authenticate the card
If Secure Channel Method = '4' or '5' (SCP02 i='55') or Secure Channel Method = '7' and Card Challenge Mode = '1' (Predictive), the following 3 fields must be present:		
AID Length	2 N	The length of the following AID field (must be even).
AID	n H	The Application Identifier used to generate a pseudo random card challenge.
Delimiter	1 A	Delimiter. Value ';'.
If Secure Channel Method = '0', '1', '4', '5', '6' or '7', the following fields must be present:		
Initial APDU Header	5 B	Initial APDU header for EXTERNAL AUTHENTICATE command (CLA, INS, P1, P2, Lc). Typically [0x80, 0x82, 0x00, 0x00, 0x10] but must be specified.
Security Level	1 B	Level of security for all secure messaging commands following the EXTERNAL AUTHENTICATE command. For Secure Channel Methods 0,1,4,5,6 (SCP02) valid values are: '0x00' : No secure messaging '0x01' : C-MAC '0x03' : Encryption and C-MAC '0x10' : R-MAC '0x11' : C-MAC and R-MAC '0x13' : Encryption, C-MAC and R-MAC For Secure Channel Method 7, (SCP03) valid values are: '0x00' : No secure messaging '0x01' : C-MAC '0x03' : Command encryption and C-MAC '0x11' : C-MAC and R-MAC '0x13' : Command encryption, C-MAC and R-MAC '0x33' : Command encryption, C-MAC, R-MAC and response encryption
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'ID'.
Error Code	2 N	'00': No error '05': Invalid Secure Channel Method '06': Invalid Security Level '07': Card Cryptogram verification error '08': CK-DEK parity error '09': Invalid card key generation mode '10': KMC, PSK or CK-MAC parity error '11': CK-ENC parity error '37': Invalid card challenge mode '80': AID length error 'E4': KMC – key block error 'E5': CK-ENC – key block error 'E6': CK-MAC – key block error 'E7': CK-DEK – key block error 'E8': PSK – key block error or a standard error code.
If Error Code = 'E4', 'E5', 'E6', 'E7' or 'E8', the following field will be present:		
Additional Error Code	2 A	The key block specific error code
If Secure Channel Method = '0', '1', '4', '5' or '6' (SCP02), the following fields will be present:		
APDU Header	5 B	APDU header for the EXTERNAL AUTHENTICATE command (CLA, INS, P1, P2, Lc).
Host Cryptogram	8 B	Host cryptogram for the EXTERNAL AUTHENTICATE command so that the card can authenticate the Host
Card Cryptogram	8 B	Cryptogram which would be generated by the card and used by the Host to authenticate the card. Only present if Secure Channel Method = '4' or '5'.
Authentication C-MAC	8 B	C-MAC for the EXTERNAL AUTHENTICATE command and for use as initial C-MAC in Prepare Secure Message for Chip Card command

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">If Secure Channel Method = '0', '1', '4' or '5', the following fields will be present:</td></tr>
<tr><td>SK-ENC</td><td></td><td>Session Key for cryptograms and encrypting card messages (APDU Data), if required.</td></tr>
<tr><td></td><td>'U' + 32 H</td><td>For a Variant LMK, the SK-ENC is encrypted under LMK 24-25/3.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a Key Block LMK, the SK-ENC will comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'47'</td><td>'T'</td><td>'B'</td></tr></table></td></tr>
<tr><td>SK-MAC</td><td></td><td>Session Key for authenticating card messages (C-MAC).</td></tr>
<tr><td></td><td>'U' + 32 H</td><td>For a Variant LMK, the SK-MAC is encrypted under LMK 24-25/4.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a Key Block LMK, the SK-MAC will comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'48'</td><td>'T'</td><td>'G'</td></tr></table></td></tr>
<tr><td>SK-DEK</td><td></td><td>Session Key for encrypting secret card data (e.g. application keys and PINs).</td></tr>
<tr><td></td><td>'U' + 32 H</td><td>For a Variant LMK, the SK-DEK is encrypted under LMK 24-25/5.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a Key Block LMK, the SK-DEK will comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'49'</td><td>'T'</td><td>'B'</td></tr></table></td></tr>
<tr><td>SK-RMAC</td><td></td><td>Only present if Security Level is 0x10, 0x11, or 0x13
Session Key for authenticating card responses (R-MAC).</td></tr>
<tr><td></td><td>'U' + 32 H</td><td>For a Variant LMK, the SK-RMAC is encrypted under LMK 26-27.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a Key Block LMK, the SK-RMAC will comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'48'</td><td>'T'</td><td>'V'</td></tr></table></td></tr>
<tr><td colspan="3">If Secure Channel Method = '6' and Card Key Generation Method = '0', the following field will be present:</td></tr>
<tr><td>PSK</td><td></td><td>The Personalisation System Key.</td></tr>
<tr><td></td><td>'U' + 32 H</td><td>For a Variant LMK, the PSK is encrypted under LMK 24-25/5.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a Key Block LMK, the SK-ENC will comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'40'</td><td>'T'</td><td>'N'</td></tr></table></td></tr>
<tr><td colspan="3">If Secure Channel Method = '2' (Mastercard PayPass MS Cards), the following field will be present:</td></tr>
<tr><td>KD-PERSO</td><td></td><td>KD Personalization Key for authenticating messages and encrypting sensitive data.</td></tr>
<tr><td></td><td>'U' + 32 H</td><td>For a Variant LMK, the KD-PERSO is encrypted under LMK 24-25/5.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a Key Block LMK, the KD-PERSO will comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'40'</td><td>'T'</td><td>'N'</td></tr></table></td></tr>
<tr><td colspan="3">If Secure Channel Method = '7' (SCP03), the following fields will be present:</td></tr>
<tr><td>APDU Header</td><td>5 B</td><td>APDU header for the EXTERNAL AUTHENTICATE command (CLA, INS, P1, P2, Lc).</td></tr>
<tr><td>Host Cryptogram</td><td>8 B</td><td>Host cryptogram for the EXTERNAL AUTHENTICATE command so that the card can authenticate the host.</td></tr>
<tr><td>Authentication C-MAC</td><td>8 B</td><td>C-MAC for the EXTERNAL AUTHENTICATE command.</td></tr>
<tr><td>C-MAC Chaining Value</td><td>16 B</td><td>The initial C-MAC value for the Prepare Secure Message command.</td></tr>
<tr><td>Card Cryptogram</td><td>8 B</td><td>Only present if Card Challenge Mode = '1'.
The generated card cryptogram.</td></tr>
<tr><td>SK-MAC</td><td>'S' + n A</td><td>Session Key for authenticating card messages (C-MAC).
The SK-MAC will comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'48'</td><td>'A'</td><td>'G'</td></tr></table></td></tr>
</table>

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">If Security Level = 0x03, 0x13 or 0x33, the following fields will be present:</td></tr>
<tr><td>SK-ENC</td><td>'S' + n A</td><td>Session Key for cryptograms and encrypting card messages.
The SK-ENC will comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'47'</td><td>'A'</td><td>'B'</td></tr></table></td></tr>
<tr><td colspan="3">If Security Level = 0x11, 0x13 or 0x33, the following fields will be present:</td></tr>
<tr><td>SK-RMAC</td><td>'S' + n A</td><td>Session Key for authenticating card responses (R-MAC).
The SK-RMAC will comply with the following:<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'48'</td><td>'A'</td><td>'V'</td></tr></table></td></tr>
<tr><td>End Message Delimiter</td><td>1 C</td><td>Present only if present in the command message. Value X'19.</td></tr>
<tr><td>Message Trailer</td><td>n A</td><td>Present only if present in the command message. Maximum length 32 characters.</td></tr>
</table>

Prepare Secure Message for Chip Card

Variant LMK ☑	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Not required**	

Function: This function prepares the personalization messages for transmission to a chip card after a secure session has been established based on the process defined for EMV Common Personalization Specification (CPS) and Global Platform (GP) Secure Channel Protocol 2 (SCP02).

The following protocols are supported:

- Secure Channel Protocol 02 (SCP02) "i" = 0x15
- Secure Channel Protocol 02 (SCP02) "i" = 0x55
- Support for the specific process for Mastercard PayPass Magnetic Stripe Cards
- Support for static personalization method where the card application is personalized with a single Personalisation Secret Key (PSK).
- Secure Channel Protocol 03 (SCP03)

Note that using SCP03 requires the use of an AES Key Block LMK.

Notes: There are two methods under EMV CPS and GP SCP02 that can be used to provide personalization data to the card application. Each method is described below.

Indirect (Explicit) Method:

Card data is prepared by the Data Preparation Process and sent to the Personalization Process for programming on a chip card. The Personalization Process decrypts (from under a KEK, or appropriate key) and re-encrypts (under the SK-DEK) any sensitive data, such as application keys and PINs, and creates the Application Protocol Data Unit (APDU) messages that are sent to the chip card. If secure messaging is required, the card messages are MAC protected (C-MAC), and optionally encrypted using the session keys generated when the secure session was established.

Direct (Implicit) Method:

This method is not supported at this time.

For the above methods, this function prepares the APDU message(s) performing translation of sensitive data from the Data Preparation Process key(s) to the chip card key, if required. Support is provided for single or multiple Data Groupings to be included in a single APDU message as well as extended APDU command data lengths. All Global

Platform commands are supported, including STORE DATA and PUT KEY. Depending on the Security Level established with the card, as specified in the parameters, the function will generate a C-MAC over the message and encrypt the APDU data elements where required. If the Security Level is set to require C-MAC or Encryption, bit 3 of the CLA will be automatically set to 1 by the function (typically yielding a byte of 0x84). The Global Platform specific C-MAC will always be generated with the logical channel numbers in the CLA byte (bits 1 and 2) set to 0 regardless of their setting in the Initial APDU Header. No modifications will be made to any other settings of the Initial APDU Header. The Global Platform specific options for C-MAC generation with encrypted ICVs and unmodified or modified APDUs are also supported.

For the Mastercard PayPass Magnetic Stripe method, the function prepares the APDU message and performs the translation of the KD-CVC3 key from the Data Preparation Process key to the chip card key. A MAC will also be generated over the message using the chip card key.

C-MAC is a MAC generated by the Host over the APDU command message.

NOTE: The Global Platform STORE DATA APDU is identified by an INS value of 0xE2 and bit 8 of the CLA byte set in the Initial APDU Header.

To support the ISO 7816 command APPEND RECORD which also uses an INS value of 0xE2, an additional optional flag 'GP Version Identifier' has been added which if set to '99' disables the checking for the GP STORE DATA command.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'IE'.
Secure Channel Method	1 N	'0': Indirect (Explicit) initiation of secure channel by the personalization system '1': Direct (Implicit) initiation of secure channel using pre-computed APDUs. Not supported at this time. '2': Mastercard PayPass Magnetic Stripe Cards *Note: For this method, the KD-PERSO key provided for encryption in the data groupings will also be used to generate the MAC. Only a single APDU will be created with a 1 byte length field (Lc).* '6': Indirect (Explicit) initiation of Open Platform secure channel by the personalization system using a single personalization key PSK. (SCP02) '7': Secure Channel Protocol 03 (SCP03)
If Secure Channel Method = '0', '6' or '7' the following field must be present:		
Security Level	1 B	Level of security for the chip card message created by this command. Must match level set with EXTERNAL AUTHENTICATE command. For Secure Channel Method 0 or 6 (SCP02) valid values are: '0x00': No secure messaging '0x01': C-MAC '0x03': Encryption and C-MAC '0x10': R-MAC

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td></td><td></td><td>'0x11': C-MAC and R-MAC
'0x13': Command encryption, C-MAC and R-MAC
For Secure Channel Method 7, (SCP03) valid values are:
'0x00': No secure messaging
'0x01': C-MAC
'0x03': Command encryption and C-MAC
'0x11': C-MAC and R-MAC
'0x13': Command encryption, C-MAC and R-MAC
'0x33': Command encryption, C-MAC, R-MAC and response encryption</td></tr>
<tr><td colspan="3">If Secure Channel Method = '0' or '6' (SCP02) and Security Level = 0x01, 0x03, 0x11 or 0x13, the following fields must be present:</td></tr>
<tr><td>SK-MAC or PSK</td><td></td><td>Session Key for authenticating messages (C-MAC) or the Personalisation Master Key.</td></tr>
<tr><td></td><td>'U' + 32 H</td><td>For a Variant LMK:
If Secure Channel Method = '0', the SK-MAC will be encrypted under Variant 4 of LMK pair 24-25.
If Secure Channel Method = '6', the PSK will be encrypted under variant 5 of LMK pair 24-25.</td></tr>
<tr><td></td><td>or
'S' + n A</td><td>For a Key Block LMK, the SK-MAC or PSK in key block format; must comply with one of the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'48' (SK-MAC)</td><td>'T'</td><td>'G'</td></tr>
<tr><td>'40' (PSK)</td><td>'T'</td><td>'N'</td></tr>
</table></td></tr>
<tr><td>Initial C-MAC</td><td>8 B</td><td>The C-MAC from the previous APDU command or from EXTERNAL AUTHENICATE for the first C-MAC</td></tr>
<tr><td>ICV Encryption Flag</td><td>1 N</td><td>Global Platform option to encrypt previous C-MAC for use as ICV
'0': Do not encrypt previous C-MAC before using it as ICV
'1': Encrypt previous C-MAC before using it as ICV</td></tr>
<tr><td>C-MAC Flag</td><td>1 N</td><td>Use Global Platform C-MAC generation process and options (previous C-MAC used as IV for MAC)
'0': Modify APDU prior to C-MAC generation (set secure messaging bit of CLA to 1 and include C-MAC length in Lc byte)
'1': Do not modify APDU prior to C-MAC generation
Use EMV CPS C-MAC generation process (previous C-MAC prepends MAC data, use IV of binary zeroes)
'9': Modify APDU for C-MAC generation (set secure messaging bit of CLA to 1 and include C-MAC length in Lc byte)</td></tr>
<tr><td colspan="3">If Secure Channel Method = '0' or '6' (SCP02) and Security Level = 0x03 or 0x13, the following field must be present:</td></tr>
<tr><td>SK-ENC</td><td></td><td>Session Key for cryptograms and encrypting card messages (APDU), if required, encrypted under the LMK.</td></tr>
<tr><td></td><td>'U' + 32 H</td><td>For a Variant LMK, the SK-ENC is encrypted under variant 3 of LMK pair 24-25.</td></tr>
<tr><td></td><td>'S' + n A</td><td>For a Key Block LMK, the SK-ENC must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'47'</td><td>'T'</td><td>'B'</td></tr>
</table></td></tr>
<tr><td colspan="3">If Secure Channel Method = '7' and Security Level is not 0x00, the following fields must be present:</td></tr>
<tr><td>SK-MAC</td><td>'S' + n A</td><td>Session Key for cryptograms and encrypting card messages (APDU), if required, encrypted under the LMK.
The SK-ENC must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'48'</td><td>'A'</td><td>'G'</td></tr>
</table></td></tr>
<tr><td>C-MAC Chaining Value</td><td>16 B</td><td>C-MAC Chaining from the previous APDU command or the initial C-MAC value returned from the EXTERNAL AUTHENICATE.</td></tr>
</table>

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td>Initial ICV Counter</td><td>6 N</td><td>The counter is set to 1 following a successful EXTERNAL AUTHENTICATE command and incremented for each subsequent APDU command with the secure session.</td></tr>
<tr><td colspan="3">If Secure Channel Method = '7' and Security Level = 0x03, 0x13 or 0x33, the following field must be present:</td></tr>
<tr><td>SK-ENC</td><td>'S' + n A</td><td>Session Key for cryptograms and encrypting card messages (APDU), if required, encrypted under the LMK.
The SK-ENC must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'47'</td><td>'A'</td><td>'B'</td></tr>
</table></td></tr>
<tr><td colspan="3">If Secure Channel Method = '0', '2', '6' or '7', the following fields must be present (as indicated):</td></tr>
<tr><td>Initial APDU Header</td><td>4 B</td><td>Initial APDU header for command [CLA, INS, P1, P2]
Note: Lc will be automatically generated by the function.
It is the responsibility of the host to specify the following:
CLA: Bit 8 set to1 to denote a Global Platform command otherwise 0 for ISO 7816
CLA: Bits 1 and 2 to denote the channel (default 00)
INS: Identifies the APDU.
P1: The value depends upon the INS value.
P2: The value depends upon the INS value.
The 'IE' command will copy the Initial APDU Header to the output APDU header and will make the following changes:
If bit 8 of CLA is set denoting Global Platform and the Security Level is either 1 or 3, the CLA bit 3 will be set to 1 otherwise 0.
The command will set bits 1 and 2 of the CLA byte to 00 for C-MAC processing however their initial values will be preserved for output.
If INS=0xE2 and bit 8 of the CLA byte is set denoting STORE DATA:
• the P2 value will be incremented for each additional APDU message created as a result of the APDU data length exceeding the maximum length of the command.
• Bits 5 & 4 of P1 should be set appropriately to match the DGI/TLV length setting as specified by the (optional) GP Version Identifier parameter later in this command (as defined in Table 11-89 STORE DATA Reference Control Parameter 1 of Global Platform Card Specification Version 2.3.1).</td></tr>
<tr><td>Command Length</td><td>1 N</td><td>Maximum length of command:
<table>
<tr><td>'0': Lc =1 byte. Maximum of 255 bytes for overall command, including APDU header.</td><td rowspan="2">Note: If the data groups provided cannot be accommodated in a single command, multiple commands will be created as required.
Secure Channel Method 2 only supports a single command.</td></tr>
<tr><td>'1': Lc =1 byte with a maximum length of 255.</td></tr>
<tr><td>'2': Lc = 2 bytes with a maximum length of 65,535.</td><td>Note: In this mode, the length will be preceded by 0x00 for a total of 3 bytes. Only a single command will be created.
Not a valid setting for Secure Channel Method 2.</td></tr>
<tr><td>'3': Lc = 1 byte with a maximum length of 255 including APDU header. Used with Method 0 and Security Level 00 or 01 or 03.</td><td>Note: If the data groups provided cannot be accommodated in a single command for '3', multiple commands will be created as required, and the DataGroup length will be coded in 3 bytes (preceded by 0xFF and followed by 2 bytes length).
For short data where no multiple commands are created, this mode will behave exactly same as mode '0'.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td>DG Count</td><td>2 N</td><td>For STORE DATA APDUs identified by INS value 0xE2 in the initial APDU Header, DG Count represents the Number of Data Groupings for the command i.e the number of individual DGIs to be concatenated into a single STORE DATA APDU command.
NOTE: STORE DATA APDUs are identified by an INS value of 0xE2 and bit 8 of the CLA byte set in the Initial APDU Header.
For all other APDUs, DG Count represents the number of data elements that are appended together to build the APDU data.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value ';'. Optional; only present if the following field is present AND the Secure Channel Method = '0' or '6' or '7'.</td></tr>
<tr><td>Output Mode</td><td>1 N</td><td>'0': pack DGIs into one APDU.
'1': wrap each DGI into a separate APDU.
'2': pack DGIs into one APDU and automatically set Last Store Data indicator (Bit 8) in P1 for last APDU generated
'3': wrap each DGI into a separate APDU and automatically set Last Store Data indicator (Bit 8) in P1 for last APDU generated
Optional; only present if the Secure Channel Method = '0'.</td></tr>
<tr><td colspan="3">The following fields must be repeated for each Data Group:</td></tr>
<tr><td>DGI</td><td>2 B</td><td>Identifier of the Data Group.
NOTE: this is only applicable when INS=0xE2 STORE DATA otherwise this field is ignored and should be set with 0x00 0x00.</td></tr>
<tr><td>DG Type Flag</td><td>1 A</td><td>Data Group data type
'0': Clear Data
'1': Key(s) encrypted under a KEK, ECB mode.
'2': Key encrypted under a KEK, CBC mode.
'3': PIN Block encrypted under a ZPK.
'4': Clear Data to be encrypted under SK-DEK. Must be a multiple of 8 bytes.
'5': The Key encrypted under the LMK.
Note: Encrypted Keys and PIN Blocks should be in format required by the card. (e.g. PIN Block Format 05). No format conversion will be performed.
If multiple keys are provided for a DG Type 1 group, they are to be concatenated without the Key Scheme indicator. All keys must be of the type and length defined by the Key Scheme (KEK) parameter.</td></tr>
<tr><td colspan="3">If DG Type Flag =' 1', '2' or '3', the following field must be present:</td></tr>
<tr><td rowspan="3">Decryption Key</td><td></td><td>Key to decrypt DG data.
Optional, only has to be provided with the first Data Group using this key if multiple Data Groups requiring the same key are provided.</td></tr>
<tr><td>'U' + 32 H
or
'T' + 48 H</td><td>For a Variant LMK:
If DG Type Flag = 1 or 2: KEK encrypted under LMK 24-25/1.
If DG Type Flag = 3: ZPK encrypted under LMK 06-07.</td></tr>
<tr><td>or
'S' + n A</td><td>For a Key Block LMK, the Decryption Key must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'54' (KEK)</td><td>'T'</td><td>'B', 'D', 'E', 'N'</td></tr>
<tr><td>'72' (ZPK)</td><td>'T'</td><td>'B', 'D', 'N'</td></tr>
</table></td></tr>
<tr><td colspan="3">If DG Type Flag =' 1', '2', '3' or '4', the following 4 fields must be present:</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Mandatory Delimiter. Value ';'.</td></tr>
<tr><td>Key Scheme (KEK)</td><td>1 A</td><td>If a Key Scheme was used to encrypt keys under KEK, it should be specified here. See the Key Scheme Table in Appendix A of the payShield 9000 Host Programmer's Manual. Default scheme 'X'.
Optional. Only used if DG Type Flag = 1 and Decryption Key is provided.
For a Key Block LMK, this field is ignored and should be set to '0'.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Mandatory Delimiter. Value ';'.</td></tr>
<tr><td>IV</td><td>8 B</td><td>Initialization Vector for CBC mode decryption.
Only present if DG Type Flag = 2.</td></tr>
<tr><td colspan="3">If Secure Channel Method = '0', '2' or '6' and DG Type Flag = '1', '2', '3', '4' or '5', the following field must</td></tr>
</table>

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">be present:</td></tr>
<tr><td rowspan="3">SK-DEK or KD-Perso or PSK (if not previously loaded)</td><td></td><td>Session Key for encrypting secret card data (e.g. application keys and PINs) encrypted under the LMK.
Optional, only has to be provided when DG Type Flag = 1, 2, 3, or 4 with the first Data Group requiring the key if multiple Data Groups requiring the key are provided.</td></tr>
<tr><td>'U' + 32 H</td><td>For a Variant LMK, the session Key is encrypted under LMK 24-25/5.</td></tr>
<tr><td>or
'S' + n A</td><td>For a Key Block LMK, the session Key must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'49' (SK-DEK)</td><td>'T'</td><td>'E', 'B'</td></tr>
<tr><td>'40' (PSK)</td><td>'T'</td><td>'N'</td></tr>
</table></td></tr>
<tr><td colspan="3">If Secure Channel Method = '7' and DG Type Flag = '1', '2', '3', '4' or '5', the following field must be present:</td></tr>
<tr><td>CK-DEK</td><td>'S' + n A</td><td>The Card key for encrypting secret card data e.g. application keys and PINs.
For a Key Block LMK, the card key must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'39'</td><td>'A'</td><td>'B'</td></tr>
</table></td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Mandatory Delimiter. Value ';'.</td></tr>
<tr><td colspan="3">If DG Type Flag = '5', the following field must be present:</td></tr>
<tr><td>Key Type Code</td><td>3 H</td><td>For Variant LMK, the field indicates the LMK pair and variant under which the key provided in the DG Data is encrypted under. Valid values are:
'30D': CK-ENC encrypted under LMK pair 36-37 variant 3
'40D': CK-MAC encrypted under LMK pair 36-37 variant 4
'50D': CK-DEK encrypted under LMK pair 36-37 variant 5
'507': PSK encrypted under LMK pair 24-25 variant 5
For Key Block LMK, this field will be ignored and should be set to 'FFF'.</td></tr>
<tr><td colspan="3">For all DG Type Flag values, the following 4 fields must be present:</td></tr>
<tr><td>DG Length</td><td>2 B</td><td>Length of the Data Group
If DG Type Flag = '5', this field will be ignored.</td></tr>
<tr><td>DG Data</td><td></td><td>Data Group Data</td></tr>
<tr><td></td><td>n B</td><td>If DG Type Flag = 0, 1, 2 or 4.</td></tr>
<tr><td></td><td>16 H</td><td>If DG Type Flag = 3, this is the PIN Block.
Note: DG Length above would be '08'.</td></tr>
<tr><td></td><td>'S' + n A</td><td>If DG Type Flag = 5, this is the key encrypted under the LMK.
For Secure Channel Methods = '0' and '6', the following keys are valid:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'37' (CK-ENC)</td><td>'T'</td><td>'X'</td></tr>
<tr><td>'38' (CK-MAC)</td><td>'T'</td><td>'X'</td></tr>
<tr><td>'39' (CK-DEK)</td><td>'T'</td><td>'X'</td></tr>
<tr><td>'40' (PSK)</td><td>'T'</td><td>'N'</td></tr>
</table>
For Secure Channel Method = '7', the following keys are valid:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'37' (CK-ENC)</td><td>'A', 'T'</td><td>'X'</td></tr>
<tr><td>'38' (CK-MAC)</td><td>'A', 'T'</td><td>'X'</td></tr>
<tr><td>'39' (CK-DEK)</td><td>'A', 'T'</td><td>'X'</td></tr>
<tr><td>'40' (PSK)</td><td>'T'</td><td>'N'</td></tr>
</table></td></tr>
<tr><td>DG End Delimiter</td><td>1 A</td><td>Mandatory Delimiter. Value ';'. (semicolon)</td></tr>
<tr><td>All DG End Delimiter</td><td>1 A</td><td>Mandatory Delimiter. Value ':'. (colon)</td></tr>
</table>

Field	Length & Type	Details
GP Version Delimiter	1 A	Value '$'. Optional; if present, the following field must be present.
GP Version Identifier	2 N	Indicates to which GP Card Specification this command must conform The valid values are: '01': Global Platform command with DGI format lengths (default). '02': Global Platform command with BER-TLV lengths. '99': ISO 7816 command with DGI format lengths.
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'IF'.
Error Code	2 N	'00': No error '05': Invalid Secure Channel Method '06': Invalid Security Level '07': Invalid DG Type Flag '08': SK-DEK parity error '09': Decryption Key parity error '10': SK-MAC or PSK parity error '11': SK-ENC parity error '50': Invalid ICV Encryption Flag '51': Invalid C-MAC Flag '52': Invalid Command Length '53': Decryption Key required and not available '54': SK-DEK required and not available '55': Length of APDU Data exceeds limit '56': Data Groupings block error (delimiter) '57': Maximum number of APDU messages exceeded '58': Invalid Key Type Code 'DA : Decryption key block error 'E1': SK-MAC key block error 'E2': SK-ENC key block error 'E3': SK-DEK or KD-PERSO key block error 'E8': PSK key block error 'F2' : Invalid GP Version Identifier 'F3': DG Type Flag '5' key block error 'F4': Key type not allowed for this secure channel or a standard error code.
If Error Code = 'DA', 'E1', 'E2', 'E3' or 'E8', the following field will be present:		
Additional Error Code	2 A	Key Block specific error code
If Secure Channel Method = '0', '6' or '7', the following 5 fields will be present:		
APDU Message Count		Number of commands generated by the function. Note: If Command Length = 2, only a single command will be returned.
	1 N	If Output Mode is '0' or '2'.
	2 N	If Output Mode is '1' or '3'.
The following 3 fields will be repeated for each APDU message:		
APDU Header n	5 or 7 B (if Command Length = 2)	APDU header for command [CLA, INS, P1, P2, Lc]. Note: If Command Length = 2, the Lc byte will be 0x00 followed by an additional 2 bytes indicating the data block length (3 bytes total)
APDU Data Block n	n B	APDU data (Length Lc) - Data Groupings and MAC, if required. Note: The last 8 bytes of the last, or only, APDU Data Block may consist of a C-MAC, if required by the function. This may be needed

Field	Length & Type	Details
		for use as the Initial C-MAC in the next Prepare Secure Message for Chip Card function. Note: If Command Length = 3, the Data Group Length will be coded in 3 bytes (0xFF followed by an additional 2 bytes indicating the data group length).
APDU End Delimiter	1 A	Delimiter. Value ';'. (semicolon)
All APDU End Delimiter	1 A	Delimiter. Value ':'. (colon)
If Secure Channel Method = '2', the following 2 fields will be present:		
APDU Header	5 B	APDU header for command [CLA, INS, P1, P2, Lc].
APDU Data Block	n B	APDU data (Length Lc)
If Secure Channel Method = '7' and Security Level is not 0x00, the following 2 fields will be present:		
C-MAC Chaining Value	16 B	The C-MAC computed on the final APDU message.
Final ICV Counter	6 N	The ICV value used on the final APDU message.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Verify and Decrypt Response Secure Message from Chip Card

Variant LMK ☒	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Not required**	

Function: Verify response R-MAC and optionally decrypt response data from APDU response

Notes: Depending on the security level defined in the initiation of the Secure Channel, all subsequent APDU responses within the Secure Channel may require secure messaging by use of an R-MAC for integrity and encryption for confidentiality. This command checks the integrity of the response command and decrypts the response data. No encryption is applied to a response where there is no response data field and in this case the message shall be protected only.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'II'.
Secure Channel Method	1 N	'7': Secure Channel Protocol 03 (SCP 03)
Security Level	1 B	Valid values ae: '0x11': C-MAC and R-MAC '0x13': Command encryption, C-MAC and R-MAC '0x33': Command encryption, C-MAC, R-MAC and response encryption
SK-RMAC	'S' + n A	Session Key for authenticating response messages (R-MAC) encrypted under the Key Block LMK which should conform to: **Key Usage** / **Algorithm** / **Mode of Use**: '48' / 'A' / 'V', 'N'
C-MAC Chaining Value	16 B	The C-MAC Chaining from the previous APDU command or from EXTERNAL AUTHENTICATE for the first C-MAC.
SK-ENC	'S' + n A	Only present if Security Level is '0x33'. Session Key for encrypting response messages encrypted under the Key Block LMK which should conform to: **Key Usage** / **Algorithm** / **Mode of Use**: '47' / 'A' / 'B', 'N'
Encryption Counter	5 N	The Encryption Counter's start value shall be set to 1 for the first command following a successful EXTERNAL AUTHENTICATE command. The Off-Card entity shall increment the Encryption Counter for each subsequent APDU command sent within the secure channel session.
Response Data Length	4 N	The length of the Response Data.
Response Data	n B	The APDU response data either plain text or encrypted according to the Security Level.
R-MAC	8 B	The R-MAC generated on the APDU Response Data.
Status Words	2 B	Status bytes returned in APDU response message.
LMK Delimiter	1 A	Value '%'. Optional; if present, the following field must be present.
LMK Identifier	2 N	LMK identifier; min value = '00'; max value is defined by License; must be present if the above Delimiter is present.

Field	Length & Type	Details
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'IJ'.
Error Code	2 N	'00': No error '05': Invalid Secure Channel Method '06': Invalid Security Level 'E2': SK-ENC – key block error 'E4': SK-RMAC – key block error 'E5': R-MAC verification error
If Error Code = 'E2' or 'E4', the following field will be present:		
Additional Error Code	2 A	Key Block specific error code
If Security Level = 0x33, the following 2 fields will be present:		
Response Data Length	4 N	The length of the plaintext Response Data field.
Response Data	n B	The decrypted plaintext Response Data.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Mobile Device Provisioning Commands

Use of these commands requires the optional HSM9-LIC018 license.

The HSM provides the following host commands to support the secure operations required for provisioning cryptographic material to a mobile device:

Validate Authentication Code

Variant LMK ☒	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Optional** Activity: **diag.host**	

Function: Validate Authentication Code.
The Authentication Code is generated by the Mobile Payment Application (MPA) and sent to the host Credentials Management System (CMS) during the initialization phase, allowing the CMS to authenticate the MPA.

Notes: The Authentication code is generated using the Session ID, the Mobile Device Identifier and MPA Fingerprint.

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'IQ'.
CMSMPA_ID	32 B	The Mobile Payment Application Identifier. Generated by the CMS and loaded on the MPA at initialization.
Encrypted Session ID LMK	32 B	The Session ID encrypted under the AES Key Block LMK.
MPA_FPG	32 B	The Mobile device finger print returned by the MPA
CMSMPA_AUTH	32 B	The Authentication Code generated by the MPA
End Message Delimiter	1 C	Must be present if a message trailer is present. Value X'19.
Message Trailer	n A	Optional. Maximum length 32 characters.

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'IR'.
Error Code	2 A	'00': No error '01': Failed to validate authentication code 'A1': Invalid LMK scheme Or a standard error code
If Error Code = '01' and the HSM is in Authorized State, the following field will be present:		
Diagnostic Data	32 B	The calculated CMSMPA_AUTH.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Generate Remote Management Secure Message

Variant LMK ☒	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Not required**	

Function: Generate Remote Management Secure Message

Notes: The Remote Management session keys are used to create a secure data block for transfer from the host Credentials Management System (CMS) to the Mobile Payment Application (MPA).

The format of the message from CMS to MPA is of the form:

Counter || Encrypted Message data || MAC

where Counter is incremented for each message.

This function allows the host to create message payloads containing both plain text and cipher text data generated by the data preparation systems into a secure message encrypted and protected using the Remote Management Session keys.

Note that the Single Use Key Block (DC_SUK) is generated using the 'IY' command.

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">COMMAND MESSAGE</td></tr>
<tr><td>Message Header</td><td>m A</td><td>Subsequently returned to the Host unchanged.</td></tr>
<tr><td>Command Code</td><td>2 A</td><td>Value 'IU'.</td></tr>
<tr><td>Mode</td><td>1 N</td><td>'0': Encrypt and MAC the payload using the remote management session keys MS_KEY_CONF, MS_KEY_MAC
'1': Encrypt the payload using AES ECB mode under the transport key.
'2': As Mode '0' but with binary key data elements converted to ASCII Hex format so that the plain text payload is suitable for interpretation by a JSON format parser.
'3': Encipher the payload with a Transport Key using CCM (Counter based with CBC MAC) encryption.
'4': Encipher the payload with the remote management master keys M_KEY_CONF using AES CBC and MAC protect with M_KEY_MAC using AES MAC Algorithm 1 in ISO/IEC 9797-1 with padding method 2.</td></tr>
<tr><td colspan="3">If Mode is '0' or '2', the following 3 fields must be present:</td></tr>
<tr><td>Message Counter</td><td>5 N</td><td>The message counter is maintained by the CMS and is incremented for each message sent from the CMS to MPA.
Maximum value is 65535.</td></tr>
<tr><td>MS_KEY_CONF</td><td>'S' + n A</td><td>The Remote Management Session Key for message confidentiality, encrypted under AES Key Block LMK and which must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'35'</td><td>'A'</td><td>'B', 'D', 'E', 'N'</td></tr>
</table></td></tr>
<tr><td>MS_KEY_MAC</td><td>'S' + n A</td><td>The Remote Management Session Key for message integrity, encrypted under AES Key Block LMK and which must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'36'</td><td>'A'</td><td>'C', 'G', 'N', 'V'</td></tr>
</table></td></tr>
<tr><td colspan="3">If Mode is '1', the following field must be present:</td></tr>
<tr><td>Transport Key</td><td>'S' + n A</td><td>The Transport Key encrypted under the AES Key Block LMK which must comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'55'</td><td>'A'</td><td>'E', 'N'</td></tr>
</table></td></tr>
<tr><td colspan="3">If Mode is '3', the following fields must be present:</td></tr>
<tr><td>Cipher Mode</td><td>1 N</td><td>Cipher used to encrypt the payload.
Valid values are:
'0': AES</td></tr>
<tr><td>Transport Key</td><td>'S' + n A</td><td>The transport key, which must comply with:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'24'</td><td>'A'</td><td>'B', 'N'</td></tr>
</table></td></tr>
<tr><td>MAC Length</td><td>2 N</td><td>The length of the MAC (TLen). Valid values are 04, 06, 08, 10, 12, 14, and 16 bytes.</td></tr>
<tr><td>Number of bytes to encode message length</td><td>1 N</td><td>The length MLen in bytes required to store the message length. The maximum length of the message is 2^{8*MLEN}. Valid values are 2 to 8 bytes. MLen = 1 is reserved.</td></tr>
<tr><td>Nonce length</td><td>2 N</td><td>The length of the Nonce field (NLen). The maximum length is 15-MLen bytes.</td></tr>
<tr><td>Nonce</td><td>n B</td><td>The Nonce value.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value ';'.</td></tr>
<tr><td>Authentication Data Length</td><td>6 N</td><td>The length of the Authentication Data field (ALen). Zero if not required.</td></tr>
<tr><td>Authentication Data</td><td>n B</td><td>The Authentication Data.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value ';'.</td></tr>
</table>

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">If Mode is '4', the following fields must be present:</td></tr>
<tr><td>Block Cipher</td><td>1 N</td><td>Cipher used to encrypt the payload.
Valid values are:
'0': AES</td></tr>
<tr><td>Block Cipher Mode</td><td>1 N</td><td>The block cipher mode used to encrypt the payload.
Valid values are:
'0': CBC</td></tr>
<tr><td>IV</td><td>32 H</td><td>Only present if Block Cipher Mode = '0'.
Initialization Vector.</td></tr>
<tr><td>MAC Algorithm</td><td>1 N</td><td>'0': CBC-MAC (AES only)</td></tr>
<tr><td>Padding Method</td><td>1 N</td><td>'0': ISO 9797 Padding Method 2 (i.e. add 0x80 and pad with 0x00).</td></tr>
<tr><td>M_KEY_CONF</td><td>'S' + n A</td><td>The Remote Management Session Key for message confidentiality, encrypted under AES Key Block LMK and which must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'33'</td><td>'A'</td><td>'X', 'N'</td></tr>
</table>
</td></tr>
<tr><td>M_KEY_MAC</td><td>'S' + n A</td><td>The Remote Management Session Key for message integrity, encrypted under AES Key Block LMK and which must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'34'</td><td>'A'</td><td>'X', 'N'</td></tr>
</table>
</td></tr>
<tr><td colspan="3">For all modes, the following field must be present:</td></tr>
<tr><td>Message Data Item Count</td><td>2 N</td><td>The number of Message Data Items to load into a single message payload.</td></tr>
</table>

<table>
<tr><th colspan="2">Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td>These fields are repeated for each Message Data Item</td><td>Message Data Item Type</td><td>2 N</td><td>Valid Message Data Item Types for Modes '0' and '3' are:
'00': Clear Data
'01': Key encrypted under a KEK, 3DES ECB mode (decrypted key added to output block as binary data.)
'02': Key encrypted under a KEK, 3DES CBC mode (decrypted key added to output block as binary data.)
'03': PIN Block encrypted under a ZPK (decrypted PIN block added to output block as ASCII hex.)
'06': Key encrypted under a KEK, AES ECB mode translated to encryption under the mobile Data encryption Key (key encrypted under mobile data encryption key added to output block as binary data)

Valid Message Data Item Types for Mode '2' are:
'00': Clear Data
'01': Key encrypted under a KEK, 3DES ECB mode (decrypted key added to output block as ASCII hex.)
'02': Key encrypted under a KEK, 3DES CBC mode (decrypted key added to output block as ASCII hex.)
'03': PIN Block encrypted under a ZPK (decrypted PIN block added to output block as ASCII hex.)
'06': Key encrypted under a KEK, AES ECB mode translated to encryption under the mobile Data encryption Key (key encrypted under mobile data encryption key added to output block as ASCII hex)

Valid Message Data Item Types for Mode '1' are:
'05': Key encrypted under the Key Block LMK
The keys types supported are:
The M_KEY_CONF and M_KEY_MAC which must conform to:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'33', '34'</td><td>'A'</td><td>'X', 'N'</td></tr>
</table>
The ZPK must conform to:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'72'</td><td>'A'</td><td>'D', 'N'</td></tr>
</table>
The DEK must conform to:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'21'</td><td>'A'</td><td>'B', 'N'</td></tr>
</table>
Valid Message Data Item Types for Mode '04' are:
'00': Plain text data
'07': Session ID encrypted under the Key Block LMK (decrypted session ID added to block as ASCII hex).</td></tr>
<tr><td></td><td>Decryption Key</td><td>'S' + n A</td><td>Key to decrypt the Message Data Item.
Not required when Message Data Item Type is '00' or '05'.
If multiple Message Data Items requiring the same key are provided, then this field must be provided with the first Message Data Item. However, it is optional for the remaining Message Data Items using the same key.
If Message Data Item Type is:
'01' or '02': KEK encrypted under the LMK:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'54'</td><td>'T'</td><td>'B', 'D', 'N'</td></tr>
</table>
'03': ZPK encrypted under the LMK
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'P0', '72'</td><td>'T'</td><td>'B', 'D', 'N'</td></tr>
</table>
'06': Transport Key encrypted under the LMK:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'21', '24', '54'</td><td>'A'</td><td>'B', 'N'</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Only present if Decryption Key is reused from previous Message Data Item.
Value ';'.</td></tr>
<tr><td>IV</td><td>8 B</td><td>Only present if Message Data Item Type = '02'.
Initialization Vector for CBC mode decryption.</td></tr>
<tr><td>Encryption Key</td><td>'S' + n A</td><td>Only present if Message Data Item Type = '06'.
The application KEK encrypted under the LMK and which must comply with the following:
<table>
<tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr>
<tr><td>'21'</td><td>'A'</td><td>'B', 'N'</td></tr>
</table></td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Only present if Encryption Key is reused from previous Message Data Item.
Value ';'.</td></tr>
<tr><td>Message Data Item Length</td><td>6 N</td><td>The length of the Message Data Item field</td></tr>
<tr><td>Message Data Item</td><td>n B</td><td>The plain text or encrypted Message Data</td></tr>
<tr><td>Message Data Item Delimiter</td><td>1 A</td><td>Mandatory Delimiter. Value ';' (semi-colon).</td></tr>
<tr><td>All Message Data Item Delimiter</td><td>1 A</td><td>Mandatory Delimiter. Value ':' (colon).</td></tr>
<tr><td>End Message Delimiter</td><td>1 C</td><td>Must be present if a message trailer is present. Value X'19.</td></tr>
<tr><td>Message Trailer</td><td>n A</td><td>Optional. Maximum length 32 characters.</td></tr>
</table>

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'IV'.
Error Code	2 A	'00': No error '05': Length of Message Data does not equal Message Data Item Length '06': Message Counter exceeds maximum value '07': Invalid Message Data Item value '08': Key parity error '09': Message Data is not a multiple of cipher block size bytes 'D1': M_KEY_CONF – key block error 'D2': M_KEY_MAC – key block error 'D3': Invalid message Data Item Length 'D4': MS_KEY_CONF or Transport Key – key block error 'D5': MS_KEY_MAC – key block error 'D7': message length exceeds maximum 'DA': Decryption Key – key block error 'DB': Message Data Item Type 5 - key block error or mobile Data Encryption Key error 'DC': Invalid Mode value 'DD': Key usage not allowed 'E1': Invalid Cipher Mode 'E2': Invalid MAC length 'E3': Invalid Number of bytes to encode message length 'E4': Invalid Nonce length 'E5': Invalid Authentication Data Length 'E7': Invalid key length 'E8': Invalid block cipher mode 'E9': Invalid MAC Algorithm 'EA': Invalid Pad Mode 'A1': Invalid LMK scheme Or a standard error code.

Field	Length & Type	Details
If Error Code ≠ '00', the following 2 fields will be present:		
Additional Error Code	2 A	The key block specific error code, or '00' if not relevant. See reference 1.
Additional Error Code 2	2 A	The number of the Message Data Item causing the error, or '00' if not relevant.
If Mode = '0' or '2', the following 4 fields will be present:		
Encrypted Message Data Length	6 N	The length of the Encrypted Message Data field.
Encrypted Message Data	n B	The Message Data encrypted under the MS_KEY_CONF session key using AES in Counter (CTR) mode.
MAC	8 B	The MAC using AES and MAC Algorithm 1 in ISO/IEC 9797-1 with padding method 2 over the Encrypted Message Data using the MS_KEY_MAC session key.
DC_CP Hash Code	32 B	The SHA 256 hash code generated over the plain text message data.
If Mode = '1', the following 2 fields will be present:		
Encrypted Message Data Length	6 N	The length of the Encrypted message Data field.
Encrypted Message Data	n B	The Message Data encrypted under the transport key using AES in ECB mode.
If Mode = '3', the following 2 fields will be present:		
Encrypted Message Data Length	6 N	The length of the Encrypted Message Data field.
Encrypted Message Data	n B	The Message Data encrypted under the Transport Key using Counter with CBC-MAC (CCM).
If Mode = '4', the following 3 fields will be present:		
Encrypted Message Data Length	6 N	The length of the Encrypted Message Data field.
Encrypted Message Data	n B	The Message Data encrypted under M_KEY_CONF using 'Block Cipher' and 'Block Cipher Mode'.
Encrypted Message MAC	8 B	MAC over the Encrypted Message Data using M_KEY_MAC and 'MAC Algorithm'.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Validate and Recover Remote Management Secure Message from the MPA

Variant LMK ☒	Key Block LMK ☑
License: **HSM9-LIC018**	
Authorization: **Not required**	

Function: Validate and Recover Remote Management Secure Message from the Mobile Payment Application (MPA).
The Remote Management session keys are used to decrypt and authenticate the remote management response data from the MPA.

This command currently returns plain text.

Notes: The format of the message from the MPA to Credentials Management System (CMS) is:

Counter || Encrypted Message data || MAC

where Counter is incremented for each message.

This command cannot be used to validate data generated by Validate Remote Management Secure Message (IU).

Field	Length & Type	Details
COMMAND MESSAGE		
Message Header	m A	Subsequently returned to the Host unchanged.
Command Code	2 A	Value 'IW'.
Version Delimiter	1 A	Value '='. Optional; if present, the following field must be present.
Version	1 N	Command Version Number. Valid values are: '1' – Decrypt remote management message using remote management session keys and AES Counter mode. Message Data Lengths specified as 4 N. '2' – Decrypt remote management message using remote management session keys and AES Counter mode. Message Data Lengths specified as 6 N. '3' – Decrypt remote management message using transport key and Counter based CBC-MAC (CCM).
If Version is '1' or '2', the following fields must be present.		
Message Counter	5 N	The message counter is maintained by the MPA and is incremented for each message from the MPA to CMS. Maximum value is 65535 or 0x00FFFF
MS_KEY_CONF	'S' + n A	The Remote Management Session Key for message confidentiality, encrypted under AES Key Block LMK and which must comply with the following: **Key Usage** | **Algorithm** | **Mode of Use** '35' | 'A' | 'B', 'D', 'E', 'N'
MS_KEY_MAC	'S' + n A	The Remote Management Session Key for message integrity, encrypted under AES Key Block LMK and which must comply with the following: **Key Usage** | **Algorithm** | **Mode of Use** '36' | 'A' | 'C', 'G', 'N', 'V'

<table>
<tr><th>Field</th><th>Length & Type</th><th>Details</th></tr>
<tr><td colspan="3">If Mode is '3', the following fields must be present:</td></tr>
<tr><td>Cipher Mode</td><td>1 N</td><td>Cipher used to encrypt the payload.
Valid values are:
'0': AES</td></tr>
<tr><td>Transport Key</td><td>'S' + n A</td><td>The transport key, which must comply with:
<table><tr><th>Key Usage</th><th>Algorithm</th><th>Mode of Use</th></tr><tr><td>'24'</td><td>'A'</td><td>'B', 'N'</td></tr></table>The key length is 128 bits.</td></tr>
<tr><td>MAC Length</td><td>2 N</td><td>The length of the MAC (TLen). Valid values are 04, 06, 08, 10, 12, 14, and 16 bytes.</td></tr>
<tr><td>Number of bytes to encode message length</td><td>1 N</td><td>The length MLen in bytes required to store the message length. The maximum length of the message is 2^{8*MLEN}. Valid values are 2 to 8 bytes. MLen = 1 is reserved.</td></tr>
<tr><td>Nonce length</td><td>2 N</td><td>The length of the Nonce field (Nlen). The length must be 15-MLen bytes.</td></tr>
<tr><td>Nonce</td><td>n B</td><td>The Nonce value.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value ';'.</td></tr>
<tr><td>Authentication Data Length</td><td>6 N</td><td>The length of the Authentication Data field (ALen). Zero if not required.</td></tr>
<tr><td>Authentication Data</td><td>n B</td><td>The Authentication Data.</td></tr>
<tr><td>Delimiter</td><td>1 A</td><td>Value ';'.</td></tr>
<tr><td colspan="3">For all versions, the following fields must be present:</td></tr>
<tr><td>Encrypted Message Data length</td><td>4 N
or
6 N</td><td>The length of the Encrypted Message Data.
If Version is not present or = '1', this field consists of 4 digits.
If Version = '2' or '3', this field consists of 6 digits.</td></tr>
<tr><td>Encrypted Message Data</td><td>n B</td><td>The encrypted message data.
If Version = '1' and '2', the Message Data is encrypted under the MS_KEY_CONF session key using AES in Counter (CTR) mode.
If Version = '3', the Message Data is encrypted under the Transport Key using CCM. (Note the MAC is included in the Encrypted Message Data).</td></tr>
<tr><td>MAC over Encrypted Message Data</td><td>8 B</td><td>The MAC using AES and MAC Algorithm 1 in ISO/IEC 9797-1 with padding method 2 over the Encrypted Message Data using the MS_KEY_MAC session key.</td></tr>
<tr><td>End Message Delimiter</td><td>1 C</td><td>Must be present if a message trailer is present. Value X'19.</td></tr>
<tr><td>Message Trailer</td><td>n A</td><td>Optional. Maximum length 32 characters.</td></tr>
</table>

Field	Length & Type	Details
RESPONSE MESSAGE		
Message Header	m A	Returned to the Host unchanged.
Response Code	2 A	Value 'IX'.
Error Code	2 A	'00': No error '01': MAC verification failed '06': Invalid message counter '80': Invalid Message Data length 'A1': Invalid LMK scheme 'D4': MS_KEY_CONF – key block error 'D5': MS_KEY_MAC – key block error 'D6': Invalid Version value 'D7': Message length exceeds 32K 'E1': Invalid cipher mode 'E2': Invalid MAC length 'E3': Invalid Number of bytes to encode message length value 'E4': Invalid Nonce length 'E5': Invalid Authentication Data length 'E6': Invalid Encrypted Message Data Length 'E7': Invalid key length Or a standard error code.
If Error Code = 'D4' or 'D5', the following field will be present:		
Additional Error Code	2 A	The key block specific error code. See reference 1.
If Error Code = '00', the following 2 fields will be present:		
Message Data Length	4 N or 6 N	The length of the Message Data field If Version is not present or = '1', this field consists of 4 digits. If Version field = '2' or '3', this field consists of 6 digits.
Message Data	n B	The plaintext Message Data.
End Message Delimiter	1 C	Present only if present in the command message. Value X'19.
Message Trailer	n A	Present only if present in the command message. Maximum length 32 characters.

Appendices

Note 1: Compress data elements by creating bytes consisting of two digits per byte (having values in the range Hex '0'–'F'). These data elements are right justified and padded with leading hex '0's if required.

Note 2: Compress data elements by creating bytes consisting of two digits per byte (having values in the range Hex '0'–'F'). These data elements are left justified and padded with trailing hexadecimal 'F's if required.

Example: A Primary Account Number (PAN) consisting of 1234567890123 will be stored in an 8 byte field as Hex '12 34 56 78 90 12 3F FF'.

Notations: N_{CA} = Length of the Certification Authority Public Key Modulus

N_I = Length of the Issuer Public Key Modulus

N_{IC} = Length of the Card (ICC) Public Key Modulus

Appendix A – Self-Signed Issuer Public Key Certificate Format (Visa)

Field Name	Length & Format	Description	
Header	1 b	Hex value '22'	NOT SIGNED
Length of Issuer Public Key Modulus	1 b	Length of Issuer Public Key Modulus in Hex. (Number of bytes)	NOT SIGNED
Issuer Public Key Modulus	var b	Issuer's Public Key Modulus	NOT SIGNED
Issuer Public Key Exponent Length	1 b	Length of Issuer Public Key Exponent. (Number of bytes). Must be either hex 01 (for exponent 3) or hex 03 (for exponent 65537).	NOT SIGNED
Issuer Public Key Exponent	1 or 3 b	Issuer Public Key Exponent. Must be either 3 or 65537, that is, hex 03 or hex 01 00 01.	NOT SIGNED
Tracking Number	3 b	Tracking number from Visa Financial Institution registration form. See Note 1.	NOT SIGNED

Self-Signed Issuer Public Key Certificate

(Clear Data – Unsigned Issuer Public Key Input Extension)

Field Name	Length & Format	Description	Hashed
Header	1 b	Hex. value '23'.	Yes
Visa Service Identifier	4 b	Identifies a Visa Service. The Proprietary Application Identifier Extension (PIX) is left justified and padded on the right with four hex zeros. Current valid International Service Identifiers are: Hex 1010 0000 : for Debit/Credit Hex 2010 0000 : for Electron Hex 3010 0000 : for Interlink Hex 8010 0000 : for PLUS Check with Visa regional offices for Regional/National service identifiers	Yes
Certificate Format	1 b	Hex value '02'.	Yes
Issuer Identification Number	4 b	Leftmost 3-8 digits from the PAN, right padded with hexadecimal Fs. See Note 2.	Yes
Certificate Expiration Date	2 b	MMYY after which this certificate is invalid. See Note 1.	Yes
Tracking Number	3 b	Tracking number from Visa Financial Institution registration form. See Note 1.	Yes
Hash Algorithm Indicator	1 b	Identifies the hash algorithm used to produce the Hash Result.	Yes
Issuer's Public Key Algorithm Indicator	1 b	Identifies the digital signature algorithm to be used with the Issuer's Public Key.	Yes
Issuer Public Key Modulus Length	1 b	Identifies the length of the Issuer Public Key Modulus. (Number of bytes)	Yes
Issuer Public Key Exponent Length	1 b	Length *e* of the Issuer Public Key Exponent (Number of bytes). Either hex 01 (for exponent 3) or hex 03 (for exponent 65537).	Yes
Leftmost Digits of Issuer Public Key Modulus	var b	Leftmost N_I - (39 + *e*) bytes of Issuer's Public Key Modulus	Yes
Issuer Public Key Exponent	1 or 3 b	Issuer Public Key Exponent. Is either 3 or 65537, that is, hex 03 or hex 01 00 01.	Yes
Hash Result	20 b	Hash of Issuer's Public Key and its related information.	No

Self-Signed Issuer Public Key Certificate

(Self-Signed Certificate Data)

Appendix B – Self-Signed Issuer Public Key Certificate Format (Mastercard)

Field Name	Length & Format	Description		Hashed
ID of Subject Certificate (Issuer Identifier)	4 b	Leftmost 3-8 digits from the PAN, right padded with hexadecimal F. See Note 2.	NOT SIGNED	No
Issuer Public Key Index	3 b	Number, chosen by the Issuer, which uniquely identifies the Public Key. See Note 1.	NOT SIGNED	No
Subject Public Key Algorithm Indicator (Signature Identifier)	1 b	Indicates the signature algorithm to be used with the Issuer Public Key.	NOT SIGNED	No
Subject Public Key Modulus Length	1 b	Length of Issuer Public Key Modulus (equal to N_I)	NOT SIGNED	No
Subject Public Key Exponent Length	1 b	Length of Issuer Public Key Exponent. (Number of bytes). Must be either hex 01 (for exponent 3) or hex 03 (for exponent 65537).	NOT SIGNED	No
Leftmost Digits of Subject Public Key Modulus	N_I-36 b	N_I-36 most significant bytes of the Issuer Public Key Modulus	NOT SIGNED	No
Subject Public Key Modulus Remainder	36 b	36 least significant bytes of the Issuer Public Key Modulus	NOT SIGNED	Yes
Subject Public Key Exponent	1 or 3 b	Issuer Public Key Exponent. Is either 3 or 65537, that is, hex 03 or hex 01 00 01.	NOT SIGNED	Yes

Self-Signed Issuer Public Key Certificate

(Clear Data)

Field Name	Length & Format	Description	Hashed
Recovered Data Header	1 b	Hex value '6A'.	No
Certificate Format	1 b	Hex value '11'.	Yes
ID of Certificate Subject (Issuer Identifier)	4 b	Leftmost 3-8 digits from the PAN, right padded with hex. 'F'. See Note 2.	Yes
Certificate Expiry Date	2 b	MMYY after which this certificate is invalid. See Note 1.	Yes
Certificate Serial Number	3 b	Chosen by the Issuer. See Note 1.	Yes
Hash Algorithm Indicator	1 b	Identifies the hash algorithm used to produce the Hash Result.	Yes
Subject Public Key Algorithm Indicator	1 b	Identifies the digital signature algorithm to be used with the Issuer's Public Key	Yes
Subject Public Key Modulus Length	1 b	Length of the Issuer Public Key Modulus in bytes (N_I)	Yes
Subject Public Key Exponent Length	1 b	Length of the Issuer Public Key Exponent (Number of bytes). Either hex 01 (for exponent 3) or hex 03 (for exponent 65537).	Yes
Leftmost Digits of Subject Public Key Modulus	N_I-36 b	Leftmost N_I-36 bytes of Issuer's Public Key Modulus	Yes
Hash Result	20 b	Hash of Issuer Public Key and its associated information.	No
Recovered Data Trailer	1 b	Hex value 'BC'.	No

Self-Signed Issuer Public Key Certificate

(Self-Signed Certificate)

Appendix C – Self-Signed Issuer Public Key Certificate Format (American Express)

Field Name	Length & Format	Description	
Header	1 b	Hex value '22'.	NOT SIGNED
Length of Issuer Public Key Modulus	1 b	Length of Issuer Public Key in Hex. (Number of bytes)	NOT SIGNED
Issuer Public Key Modulus	var b	Issuer's Public Key Modulus	NOT SIGNED
Issuer Public Key Exponent Length	1 b	Length of Issuer Public Key Exponent. (Number of bytes). Must be either hex 01 (for exponent 3) or hex 03 (for exponent 65537).	NOT SIGNED
Issuer Public Key Exponent	1 or 3 b	Issuer Public Key Exponent. Must be either 3 or 65537, that is, hex 03 or hex 01 00 01.	NOT SIGNED
Tracking Number	3 b	Number for transmittal tracking. See Note 1.	NOT SIGNED

Self-Signed Issuer Public Key Certificate

(Clear Data – Unsigned Issuer Public Key Input Extension)

Field Name	Length & Format	Description	Hashed
Header	1 b	Hex. value '23'.	Yes
Service Identifier	4 b	American Express Product Identifier Fixed value '00 00 00 00'	Yes
Certificate Format	1 b	Hex value '02'.	Yes
Issuer Identifier (BIN)	4 b	Issuer Identification Number (left justified and padded on the right with hexadecimal 'F's). See Note 2.	Yes
Certificate Expiration Date	2 b	MMYY after which this certificate is invalid. See Note 1.	Yes
Tracking Number	3 b	Number for transmittal tracking. See Note 1	Yes

Hash Algorithm Indicator	1 b	Identifies the hash algorithm used to produce the Hash Result.	Yes
Issuer's Public Key Algorithm Indicator	1 b	Identifies the digital signature algorithm to be used with the Issuer's Public Key.	Yes
Issuer Public Key Modulus Length	1 b	Identifies the length of the Issuer Public Key Modulus. (Number of bytes)	Yes
Issuer Public Key Exponent Length	1 b	Length *e* of the Issuer Public Key Exponent (Number of bytes). Either hex 01 (for exponent 3) or hex 03 (for exponent 65537).	Yes
Most Significant Part of Issuer Public Key Modulus	var b	Leftmost N_I - (39 + *e*) bytes of Issuers Public Key Modulus	Yes
Issuer Public Key Exponent	1 or 3 b	Issuer Public Key Exponent. Is either 3 or 65537, that is, hex 03 or hex 01 00 01.	Yes
Hash Result	20 b	Hash of indicated fields.	No

Self-Signed Issuer Public Key Certificate

(Self-Signed Issuer Public Key Data)

Appendix D – Issuer Public Key Certificate Format (Visa)

Field Name	Length & Format	Description		Hashed
Header	1 b	Hex. value '24'.	NOT SIGNED	No
Visa Service Identifier	4 b	Identifies a Visa Service. The Proprietary Application Identifier Extension (PIC) is left justified and padded on the right with four hex zeros. Current valid International Service Identifiers are: Hex 1010 0000 : for Debit/Credit Hex 2010 0000 : for Electron Hex 3010 0000 : for Interlink Hex 8010 0000 : for PLUS Check with Visa regional offices for Regional/National service identifiers	NOT SIGNED	No
Issuer Identification Number	4 b	Leftmost 3-8 digits from the PAN, right padded with hexadecimal Fs. See Note 2.	NOT SIGNED	No
Certificate Serial Number	3 b	Certificate Serial Number assigned by Visa CA	NOT SIGNED	No
Certificate Expiration Date	2 b	MMYY after which this certificate is invalid. See Note 1.	NOT SIGNED	No
Issuer Public Key Modulus Remainder Length	1 b	Length of Issuer Public Key Modulus (N) Remainder in hex (Number of bytes)	NOT SIGNED	No
Issuer Public Key Modulus (N) Remainder	var b	Field only present if NI > NCA - 36, and consists of the NI-NCA + 36 least significant bytes of the Issuer Public Key Modulus (N). NI is the length, in bytes, of the Issuer Public Key Modulus and NCA is the length, in bytes, of the VSDC CA Key used for create the IPK certificate.	NOT SIGNED	Yes
Issuer Public Key Exponent Length	1 b	Length of Issuer Public Key Exponent e in hex. (Number of bytes). Either hex 01 (for exponent 3) or hex 03 (for exponent 65537).	NOT SIGNED	No
Issuer Public Key Exponent	1 or 3 b	Issuer Public Key Exponent. Is either 3 or 65537, that is, hex 03 or hex 01 00 01.	NOT SIGNED	Yes
CA Public Key Index	1 b	Public Key Index for CA Public Key used to create the Issuer Public Key Certificate	NOT SIGNED	No

Issuer Certificate

(Unsigned Data - Unsigned Issuer Public Key Output Extension)

Field Name	Length & Format	Description	Hashed
Recovered Data Header	1 b	Hex. value '6A'.	No
Certificate Format	1 b	Hex. value '02'.	Yes
Issuer Identification Number	4 b	Leftmost 3-8 digits from the PAN, right padded with hexadecimal Fs. See Note 2.	Yes
Certificate Expiration Date	2 b	MMYY after which this certificate is invalid. See Note 1.	Yes
Certificate Serial Number	3 b	Binary number unique to this certificate assigned by the Certification Authority	Yes
Hash Algorithm Indicator	1 b	Identifies the hash algorithm used to produce the Hash Result in the digital signature scheme.	Yes
Issuer Public Key Algorithm Indicator	1 b	Identifies the digital signature algorithm to be used with the Issuer Public Key.	Yes
Issuer Public Key Modulus Length	1 b	Identifies the length of the Issuer Public Key Modulus (Number of bytes)	Yes
Issuer Public Key Exponent Length	1 b	Identifies the length of the Issuer Public Key Exponent (Number of bytes)	Yes
Issuer Public Key Modulus (N) or Leftmost portion of the Issuer Public Key Modulus (N)	var b	If $N_I \leq N_{CA} - 36$, this field consists of the full Issuer Public Key Modulus (N) right padded with $N_{CA} - 36 - N_I$ 'BB' bytes. If $N_I > N_{CA} - 36$, this field consists of the $N_{CA} - 36$ most significant bytes of the Issuer Public Key Modulus (N).	Yes
Hash Result	20 b	Hash of the Issuer Public Key and related information.	No
Recovered Data Trailer	1 b	Hex. value 'BC'.	No

Issuer Certificate

(Issuer Public Key Certificate)

Field Name	Length & Format	Description
Header	1 b	Hex. value '00'.
Block Format Code	1 b	Hex. value '01'.
Padding Characters	var b	Hex. value 'FF'. The length of the padding is equal to the Signing key modulus - 38.
Separator	1 b	Hex. value '00'.
Algorithm Indicator	15 b	Hash Algorithm indicator used by the CA For SHA-1, Hex Value = '3021300906052b0e03021a05000414'
Hash Results	20 b	SHA-1 Hash of the concatenation of the Unsigned Data and the Issuer Public Key Certificate

Issuer Certificate

(Detached Signature)

Appendix E – Issuer Public Key Certificate Format (Mastercard)

Field Name	Length & Format	Description		Hashed
Issuer Identification Number	4 b	Leftmost 3-8 digits from the PAN, right padded with hexadecimal Fs. See Note 2.	NOT SIGNED	No
Issuer Public Key Index	3 b	Number, chosen by the Issuer, which uniquely identifies the Public Key.	NOT SIGNED	No
CA Public Key Index	1 b	The CA Public Key Index uniquely identifies a CA Public Key.	NOT SIGNED	No
Issuer (Subject) Public Key Modulus Remainder	var b	Field only present if $N_I > N_{CA} - 36$, and consists of the $N_I - N_{CA} + 36$ least significant bytes of the Issuer Public Key Modulus	NOT SIGNED	Yes
Issuer (Subject) Public Key Exponent	var b	Exponent (e = 1 to $N_i / 4$).	NOT SIGNED	Yes

Issuer Certificate

(Unsigned Data)

Field Name	Length & Format	Description	Hashed
Recovered Data Header	1 b	Hex. value '6A'.	No
Certificate Format	1 b	Hex. value '02'.	Yes
Issuer Identification Number	4 b	Leftmost 3-8 digits from the PAN, right padded with hexadecimal F. See Note 2.	Yes
Certificate Expiration Date	2 b	MMYY after which this certificate is invalid. See Note 1.	Yes
Certificate Serial Number	3 b	Binary number unique to this certificate assigned by the Certification Authority	Yes
Hash Algorithm Indicator	1 b	Identifies the hash algorithm used to produce the Hash Result in the digital signature scheme.	Yes
Issuer Public Key Algorithm Indicator	1 b	Identifies the digital signature algorithm to be used with the Issuer Public Key.	Yes
Issuer Public Key Modulus Length	1 b	Identifies the length of the Issuer Public Key Modulus (Number of bytes).	Yes
Issuer Public Key Exponent Length	1 b	Identifies the length of the Issuer Public Key Exponent (Number of bytes).	Yes
Issuer Public Key Modulus (N) or Leftmost portion of the Issuer Public Key Modulus (N)	var b	If $N_I \leq N_{CA} - 36$, this field consists of the full Issuer Public Key Modulus (N) right padded with $N_{CA} - 36 - N_I$ 'BB' bytes. If $N_I > N_{CA} - 36$, this field consists of the $N_{CA} - 36$ most significant bytes of the Issuer Public Key Modulus (N).	Yes
Hash Result	20 b	Hash of the Issuer Public Key and related information.	No
Recovered Data Trailer	1 b	Hex. value 'BC'.	No

Issuer Certificate

(Issuer Public Key Certificate)

Appendix F – Issuer Public Key Certificate Format (American Express)

Field Name	Length & Format	Description		Hashed
Header	1 b	Hex. value '24'.	NOT SIGNED	No
Service Identifier	4 b	American Express Product Identifier Fixed value '00 00 00 00'.	NOT SIGNED	No
Issuer Identification Number	4 b	Issuer Identification Number (left justified and padded on the right with hexadecimal 'F's). See Note 2.	NOT SIGNED	No
Certificate Serial Number	3 b	'01nnnn' where nnnn is the Tracking Number from the certificate request. See Note 1.	NOT SIGNED	No
Certificate Expiration Date	2 b	MMYY after which this certificate is invalid. See Note 1.	NOT SIGNED	No
Issuer Public Key Modulus Remainder Length (LN)	1 b	Number of bytes in Public Key Modulus Remainder (LN^R_I = max (LN_I-LN_{CA}+36, 0))	NOT SIGNED	No
Issuer Public Key Modulus Remainder (N^R_I)	var b	Field present if LN^R_I l > 0, contains LN^R_I least significant (rightmost) bytes of the Issuer Public Key Modulus	NOT SIGNED	Yes
Issuer Public Key Exponent Length (Le_I)	1 b	Number of bytes for Public Key Exponent in hex.	NOT SIGNED	No
Issuer Public Key Exponent (e_I)	Le_I b	Issuer Public Key Exponent in hex (e_I = 1 to NI / 4)	NOT SIGNED	Yes
CA Public Key Index	1 b	Public Key Index for CA Public Key used to create the Issuer Public Key Certificate.	NOT SIGNED	No

Issuer Certificate

(Unsigned Data - Unsigned Issuer Public Key Output Extension)

Field Name	Length & Format	Description	Hashed
Recovered Data Header	1 b	Hex. value '6A'.	No
Certificate Format	1 b	Hex. value '02'.	Yes
Issuer Identification Number	4 b	Issuer Identification Number (left justified and padded on the right with hexadecimal 'F's). See Note 2.	Yes
Certificate Expiration Date	2 b	MMYY after which this certificate is invalid. See Note 1.	Yes
Certificate Serial Number	3 b	'01nnnn' where nnnn is the Tracking Number from the certificate request.	Yes
Hash Algorithm Indicator	1 b	Identifies the hash algorithm used to produce the Hash Result.	Yes
Issuer Public Key Algorithm Indicator	1 b	Identifies the digital signature algorithm to be used with the Issuer's Public Key.	Yes
Issuer Public Key Modulus Length (LN_I)	1 b	Identifies the length of the Issuer Public Key Modulus (Number of bytes)	Yes
Issuer Public Key Exponent Length (Le_I)	1 b	Identifies the length of the Issuer Public Key Exponent (Number of bytes)	Yes
Issuer Public Key Modulus (N) or Leftmost portion of the Issuer Public Key Modulus (N)	LN_{CA36} b	$LN_I > LN_{CA}-36$; MS part of Modulus $LN_I = LN_{CA}-36$; Full Modulus only $LN_I < LN_{CA}-36$; Full Modulus appended with 'BB' pad of LN_{CA} -36- LN_I bytes	Yes
Hash Result	20 b	Hash of the Issuer Public Key and related information.	No
Recovered Data Trailer	1 b	Hex. value 'BC'.	No

Issuer Certificate

(Issuer Public Key Certificate)

Field Name	Length & Format	Description
Header	1 b	Hex. value '00'.
Block Format Code	1 b	Hex. value '01'.
Padding Characters	var b	Hex. value 'FF'. The length of the padding is equal to the Signing key modulus - 38.
Separator	1 b	Hex. value '00'.
Algorithm Indicator	15 b	OID for SHA-1, '30 21 30 09 06 05 2B 0E 03 02 1A 05 00 04 14'.
Hash Results	20 b	SHA-1 Hash of the concatenation of the Unsigned Data and the Issuer Public Key Certificate.

Issuer Certificate

(Detached Signature)

Appendix G – Format of Card (ICC) Public Key Certificate

Field Name	Length & Format	Description
Certificate Format	1 b	Hex value '04'.
Application PAN	10 b	PAN padded on the right with hex Fs. See Note 2.
Certificate Expiration date	2 b	MMYY, after which this certificate is invalid. See Note 1.
Certificate Serial Number	3 b	Binary number unique to this certificate assigned by the issuer. See Note 1.
Hash Algorithm Indicator	1 b	Identifies the hash algorithm used to produce the Hash Result. '01' means SHA-1.
ICC Public Key Algorithm indicator	1 b	Identifies the digital signature algorithm to be used with the ICC Public Key. '01' means RSA.
ICC Public Key Modulus length	1 b	Identifies the length of ICC Public Key Modulus in bytes
ICC Public Key Exponent length	1 b	Identifies the length of the ICC Public Key Exponent in bytes. Either hex 01 (for exponent 3) or hex 03 (for exponent 65537).
Leftmost bytes of the ICC Key Modulus	N_I-42 b	If N_{IC} =<N_I-42, this field consists of the full ICC Public Key padded to the right with N_I-42-N_{IC} bytes of value hex 'BB' if necessary. If N_{IC}>N_I-42, this field consists of the N_I-42 most significant bytes of the ICC Public Key Modulus.
ICC Public Key Modulus Remainder	0 or N_{IC}-N_I+42 b	This field is only present if N_{IC}>N_I-42 and consists of N_{IC}- N_I+42 least significant bytes of the ICC Public Key Modulus.
ICC Public Key Exponent	1 or 3 b	ICC Public Key Exponent. Is either 3 or 65537, that is, hex 03 or hex 01 00 01.
Static Data	var b	Static Data to be Authenticated. *Note: This field is only present for the Card Public Key Data and is not present for the Card PIN Encipherment Public Key Data.*

Card (ICC) Public Key Data to be signed by the Issuer

(Input to hash algorithm)

Field Name	Length & Format	Description
Recovered Data Header	1 b	Hex. value '6A'.
Certificate Format	1 b	Hex. value '04'.
Application PAN	10 b	PAN padded on the right with hex Fs. See Note 2.
Certificate Expiration date	2 b	MMYY, after which this certificate is invalid. See Note 1.
Certificate Serial Number	3 b	Binary number unique to this certificate assigned by the issuer. See Note 1.
Hash Algorithm Indicator	1 b	Identifies the hash algorithm used to produce the Hash Result below. '01' means SHA-1.
ICC Public Key Algorithm indicator	1 b	Identifies the digital signature algorithm to be used with the ICC Public Key. '01' means RSA
ICC Public Key Modulus length	1 b	Identifies the length of ICC Public Key Modulus in bytes
ICC Public Key Exponent Length	1 b	Identifies the length of the ICC Public Key Exponent in bytes. Either hex 01 (for exponent 2 or 3) or hex 03 (for exponent 65537).
ICC Public Key modulus or leftmost bytes of the ICC Public Key modulus	N_I-42 b	If N_{IC} =<N_I-42, this field consists of the full ICC Public Key padded to the right with N_I-42-N_{IC} bytes of value hex 'BB' if necessary. If N_{IC}>N_I-42, this field consists of the N_I-42 most significant bytes of the ICC Public Key Modulus.
Hash Result	20 b	Hash of Card (ICC) Public Key and its related information
Recovered data trailer	1 b	'BC'.

Content of the Card (ICC) Certificate

(Deciphered)

Appendix H – Private Key Encodings

ASN.1 encoding of a PRIVATE Key

An RSA Private Key has the following ASN.1 encoded format:

RSAPrivateKey::= SEQUENCE{

p BIT STRING,

q BIT STRING,

d1 BIT STRING,

d2 BIT STRING,

q-1 mod p BIT STRING}

When using ASN.1 encoding, the value 30 indicates a sequence and the value 03 indicates a bit string. As a result, a ASN.1 encoded private key will appear as follows:

30 | Length of Complete Sequence | 03 | Bit String Length | p | 03 | Bit String Length | q | 03 | Bit String Length | d1| 03 | Bit String Length | d2| 03 | Bit String Length | q-1 mod p |

When defining the length of the bit string, the length of the bit string in bytes is given first, followed by the number of bits that are ignored in the last byte. For example "101010" is encoded as 03 02 02 A8, where 03 is a bit string, 02 is the length, 02 is the number of bits dropped from the end of the data, and A8 is the zero padded data (note that the padding can be any value as it is ignored).

The following example shows the five data items that make up a key and then shows the key ASN.1 encoded:

p:

FADD62A62492706C 5784790CDC40D76C

5CA0736FA0E07CAA EB1729C1C7FF18E1

70EFC25B7711C907 B515542ACFD80823

q:

EC43DD6A0F955408 09579E9A8D0DECC3

B4050712A28C97F0 6521505342D6E102

58F3BBBB845CBAB0 3B136EC6A7E1F6E9

dp (d1):

A73E41C41861A048 3A5850B33D808F9D

9315A24A6B40531C 9CBA1BD68554BB40

F5F52C3CFA0BDB5A 78B8E2C7353AB017

dq (d2):

9D82939C0A638D5A B0E5146708B3F32D

22AE04B71708654A EE16358CD739EB56

E5F7D27D02E87C75 7CB79F2F1A96A49B

U (q inverse mod p):

CEB3DA4206C267C1 1EF3DCCB77268707

09E735BED60E68D5 3C0E573FB64A634F

376B15CCC0219C5A 02F09B834048ECB9

For the ASN.1 Encoded Private Key, the bit string indicators have been underlined and the length indicators are in italic to aid with clarity.

30 81 FF 03 *31 00* FA DD 62 A6 24 92 70 6C 57 84

79 0C DC 40 D7 6C 5C A0 73 6F A0 E0 7C AA EB 17

29 C1 C7 FF 18 E1 70 EF C2 5B 77 11 C9 07 B5 15

54 2A CF D8 08 23 03 *31 00* EC 43 DD 6A 0F 95 54

08 09 57 9E 9A 8D 0D EC C3 B4 05 07 12 A2 8C 97

F0 65 21 50 53 42 D6 E1 02 58 F3 BB BB 84 5C BA
B0 3B 13 6E C6 A7 E1 F6 E9 03 *31 00* A7 3E 41 C4

18 61 A0 48 3A 58 50 B3 3D 80 8F 9D 93 15 A2 4A
6B 40 53 1C 9C BA 1B D6 85 54 BB 40 F5 F5 2C 3C

FA 0B DB 5A 78 B8 E2 C7 35 3A B0 17 03 *31 00* 9D

82 93 9C 0A 63 8D 5A B0 E5 14 67 08 B3 F3 2D 22

AE 04 B7 17 08 65 4A EE 16 35 8C D7 39 EB 56 E5

F7 D2 7D 02 E8 7C 75 7C B7 9F 2F 1A 96 A4 9B 03

31 00 CE B3 DA 42 06 C2 67 C1 1E F3 DC CB 77 26

87 07 09 E7 35 BE D6 0E 68 D5 3C 0E 57 3F B6 4A

63 4F 37 6B 15 CC C0 21 9C 5A 02 F0 9B 83 40 48

EC B9

CRT Encoding of Private Key Components

This section describes how the 5 Chinese Remainder Theorem (CRT) components are formatted.

The length in bits of each of the CRT components is half the modulus length rounded up to the next bit (i.e. if the modulus is 1007 bits, then the CRT component size is 504 bits).

The size of each component is determined by defining a function CEIL(X) as X/24 rounded up to the next highest integer, and then multiplied by 3, where X is the CRT component size in bits (i.e. CEIL (504) is 504/24=21, multiplied by 3 to get 63). This will be the native size of CRT components before any formatting or padding (required to make it a multiple of 8 bytes for DES encryption).

The plaintext component block is generally formatted as follows:

Length	CRT component	Padding
(1 byte) optional	Var.	'0000..' to 8 bytes or '80' '00..' to 8 bytes or 4 bytes (8000 0000) or 8 bytes (8000 0000 0000 0000)

- If Length Bytes is 0 the length is not included with the CRT component.
- If mandatory padding ***is not*** required, the padding characters are '00' to make the length a multiple of 8 bytes.
- If mandatory padding ***is*** required, the padding characters will be a single byte of 80 followed by 00 bytes to make a multiple of 8 bytes, or 4 bytes (8000 0000) or 8 bytes (8000 0000 0000 0000).

Alternative CRT Output formats for a Private Key

Commands in this document have the ability to output a Private Key in the form of 5 Chinese Remainder Theorem components. These are p, q, d1, d2, q-1 mod p. It is also possible to select either of the conditions q>p or p>q.

Some applications may require that the 5th component (q-1 mod p) is provided in a different form such as the modular inverse of p (or p-1 mod q). It is possible to obtain an output in this format as follows:

1. If the condition q>p is required:

   - Select the condition p>q (i.e. the opposite of what is required)
   - From the returned 5 components (p, q, d1, d2, q-1 mod p) rearrange them according to the following table.

For new component	Use returned component
p	q
q	p
d1	d2
d2	d1
p^{-1} mod q	q^{-1} mod p

2. If the condition p>q is required:

   - Select the condition q>p (i.e. the opposite of what is required)
   - From the returned 5 components (p, q, d1, d2, q-1 mod p) rearrange them according to the table above.

Exponent/Modulus Encoding of Private Keys

The private key exponent (d) and modulus (n) are given in the following format:

Length | Private Key Exponent (d) or Modulus (n) | Padding

The length is an n byte value (though usually 0 or 1) which indicates the length (in bytes) of the following field, which may be the Private Key Exponent (d) or the Modulus (n). This value is given in HEX E.g. 0x40 corresponds to a key size of 64. If the number of bytes specified for the length is zero then this field will be omitted, giving an output as follows:

Private Key Exponent (d) or Modulus (n) | Padding

If mandatory padding ***is not*** required, the padding characters are '00' to make the total length of all three (or two) parts a multiple of 8 bytes.

If mandatory padding is required, the padding characters will be either 4 bytes (8000 0000) or 8 bytes (8000 0000 0000 0000).

Appendix I – MULTOS Card Public Key Certificate Format

MULTOS V3.0

A MULTOS v3.0 public key certificate (MCD_PK_C) is a total of 136 bytes, comprising:

Certificate public key length	2 bytes
Certificate key header	38 bytes
Key Certificate	96 bytes

The 38 byte Certificate key header has the following format:

Miscellaneous data	18 bytes
Public key exponent length	2 bytes
Public exponent	4 bytes
Miscellaneous data	5 bytes
msm_controls_data_date	1 byte
mcd_no	8 bytes

Notes:

- The miscellaneous data will be ignored.
- The "Public Key Exponent Length" denotes (in bytes) the actual length of the public exponent.
- The "Public Exponent" is left justified and padded with 00 to a total of 4 bytes.

Examples:

- If public exponent = 3 (decimal) then:

  "public exponent" = 03 00 00 00 , and

  "public key exponent length" = 00 01

- If public exponent = 65537 (decimal) then:

  "public exponent" = 01 00 01 00 , and

  "public key exponent length" = 00 03

Extraction of Public Key from Certificate - The Key Certificate is encrypted using the MULTOS CA Key Transport Key (TKCK_PK). The plain value of the Key Certificate comprises:

Hash Digest	16 bytes
Card public key modulus	72 bytes
Random padding	8 bytes

The Hash Digest is an Asymmetrical Hash of the certificate header.

Certificate Authentication - In order to authenticate the certificate, the hash digest recovered from the certificate must match the digest of the header. The hash algorithm is described in Appendix M. For MULTOS v3.0 it is a single hash.

MULTOS V4.0

A MULTOS v4.0 public key certificate (MCD_PK_C) can take one of two forms, depending on the relative lengths of the smart card public key modulus and the MULTOS CA Key Transport Key (TKCK_PK) modulus.

Let N = length (in bytes) of the MULTOS CA Key Transport Key (TKCK_PK) modulus and let M = length (in bytes) of the smart card public key modulus. The case of $N \geq M+56$ is not allowed. If the keys submitted in the command satisfy this condition an error code will be returned to the host.

Case 1 **(M+32) ≤ N < (M+56)**

In this case the card public key certificate (MCD_PK_C) has the following format:

Certificate public key length	2 bytes
Key Header	38 bytes
Key Certificate	N bytes

The 38 byte Key Header has the following format:

Miscellaneous data	13 bytes
Public key length	2 bytes
Certifying key length	2 bytes
Miscellaneous data	1 byte
Public key exponent length	2 bytes
Public exponent	4 bytes
Miscellaneous data	5 bytes
msm_controls_data_date	1 byte
mcd_no	8 bytes

Notes:

- The 19 bytes of miscellaneous data will be ignored by the HSM.
- The "Public exponent" is left justified and padded with 00 to a total of 4 bytes.
- The "Public key exponent length" denotes (in bytes) the actual length of the public exponent.

Examples:

- If public exponent = 3 (decimal) then:

  "Public exponent" = 03 00 00 00 , and

  "Public key exponent length" = 00 01

- If public exponent = 65537 (decimal) then:

  "Public exponent" = 01 00 01 00 , and

  "Public key exponent length" = 00 03

- If public exponent = 257 (decimal) then:

  "Public exponent" = 01 01 00 00 , and

  "Public key exponent length" = 00 02

Extraction of Public Key from Certificate - The Key Certificate is encrypted using the MULTOS CA Key Transport Key (TKCK_PK). The plain value of the Key Certificate comprises:

Hash Result	16 bytes
Padding	N-M-32 bytes
Card public key modulus	M bytes
Redundancy	16 bytes

Case 2 N < (M+32)

In this case the card public key certificate (MCD_PK_C) has the following format:

Certificate public key length	2 bytes
Key Header	38 bytes
Card public key modulus left part	M-N+32 bytes
Key Certificate	N bytes

The 38 byte Key header has the same format as in Case 1.

Extraction of Public Key from Certificate - The Key Certificate is encrypted using the MULTOS CA Key Transport Key (TKCK_PK). The plain value of the Key Certificate comprises:

Hash Digest	16 bytes
Card public key modulus right part	N-32 bytes
Redundancy	16 bytes

Concatenate the Card Public Key modulus left part and the Card Public key right part to form the card public key modulus.

Certificate Authentication - In order to authenticate the certificate, the hash digest recovered from the certificate must match the digest of the header. The hash algorithm is described in Appendix M. For a MULTOS v4.0 this is done with two asymmetric hashes.

Appendix J – MULTOS Transport Key Certifying Key File Format

The MULTOS CA Public Key (TKCK) is supplied in the following file format prior to being reformatted into standard ASN.1 DER encoded public key.

Data Field	Description	Length (in bytes)
File_Type_Code	ASCII, 4 Character. Set to "TKCK"	4
File_Protection _Method	Binary. Set to 0x01	1
File_Structure_Method	Binary. Set to 0x01	1
Consignment_File_ID	ASCII. 8 Characters. Set to "TKCK", followed by 4 characters presenting the identifier (in hex). This will be the same as the MKD_Cert_Method_ID. For example: "TKCK0113" would be version 19 (decimal) of a MULTOS 4 96 byte TKCK	8
Date	Date	4
Time	Time	3
MKD_Cert_Method_ID	Binary. Comprised of: scheme ID, 1 byte + key version number, 1 byte Currently defined scheme IDs are: 0x00 for MULTOS 3 platforms with a 96 byte TKCK 0x01 for MULTOS 4 platforms with a 96 byte TKCK 0x02 for MULTOS 4 platforms with a 128 byte TKCK	2
Key_Length	Binary. This should match the value inferred from the scheme ID byte of the MKD_Cert_Method_ID	2
Key_Data	Binary. The actual public key (TKCK_PK)	Key-Length
Hash_Code	Binary. A SHA-1 hash of the MKD_Cert_Method_ID, Key_Length and Key_Data.	20

Key data contains the public key modulus. The exponent is always assumed to be 3.

Appendix K – MULTOS Hash Modulus File Format

The MULTOS Hash Modulus File (HASH) is supplied in the following format prior to being reformatted into standard ASN.1 DER encoded format public key.

Data Field	Description	Length (in bytes)
File_Type_Code	ASCII, 4 Character. Set to "HASH"	4
File_Protection_Method	Binary. Set to 0x01	1
File_Structure_Method	Binary. Set to 0x01	1
Consignment_File_ID	ASCII. 8 Characters. Set to "HASH", followed by 4 characters presenting the identifier (in hex). This will be the same as the Hash_Method_ID. For example: "HASH0105"	8
Date	Date	4
Time	Time	3
Hash_Method_ID	Binary	2
Key_Length	Binary	2
Key_Data	Binary. The actual Hash Modulus	Key_Length
Hash_Code	Binary. A SHA-1 hash of the Hash_Method_ID, Key_Length and Key_Data.	20

Appendix L – Self Signed CA Public Key Certificate Format (Visa)

Field Name	Length & Format	Description		Hashed
Header	1 b	Hex value '20'	NOT SIGNED	No
Service Identifier	4 b	Identifies a Visa Service. The Proprietary Application Identifier Extension (PIX) is left justified and padded on the right with four hex zeros. Current valid International Service Identifiers are: Hex 1010 0000 – for Debit/Credit Hex 2010 0000 – for Electron Hex 3010 0000 – for Interlink Hex 8010 0000 – for PLUS Note: The Service Identifier of the VSDC CA Public Key is always **1010 0000**, regardless of the Service Identifier of the Issuer's Public Key as requested in the Issuer's Public Key Input File.	NOT SIGNED	No
Length of Public Key Modulus (N_{CA})	2 b	Length of CA Public Key Modulus in Hex. (No. of bytes)	NOT SIGNED	No
Public Key Algorithm Indicator	1 b	Identifies cryptographic algorithm to be used with the CA Public Key. RSA = "01" hex	NOT SIGNED	No
Public Key Exponent Length	1 b	Length of CA Public Key Exponent in Hex. (No. of bytes)	NOT SIGNED	No
Registered Application Provider Identifier (RID)	5 b	Visa Identifier = 'A000000003'	NOT SIGNED	Yes
Public Key Index	1 b	Unique CA Public Key Serial No.	NOT SIGNED	Yes
Public Key Modulus (N_{CA})	var b	CA Public Key Modulus	NOT SIGNED	Yes
Public Key Exponent (e)	var b	CA Public Key Exponent	NOT SIGNED	Yes
Hash Result	20 b	Hash of fields indicated above	NOT SIGNED	No

Self Signed CA Public Key Certificate

(Clear Data - Unsigned Output Extension)

Field Name	Length & Format	Description
Header	1 b	Hex value '21'
Service Identifier	4 b	Identifies a Visa Service. The Proprietary Application Identifier Extension (PIX) is left justified and padded on the right with four hex zeros. Current valid International Service Identifiers are: Hex 1010 0000 – for Debit/Credit Hex 2010 0000 – for Electron Hex 3010 0000 – for Interlink Hex 8010 0000 – for PLUS
Registered Application Provider Identifier (RID)	5 b	Visa Identifier = 'A000000003'
Public Key Index	1 b	Unique CA Public Key Serial Number
Certificate Expiration Date	2 b	MMYY after which this certificate is invalid. See Note 1.
Public Key Algorithm Indicator	1 b	Identifies cryptographic algorithm to be used with the CA Public Key
Leftmost portion of CA Public Key Modulus (N_{CA})	var b	(N - [36+*e*]) bytes of the CA Public Key Modulus (N)
Hash Algorithm Indicator	1 b	Identifies the hash algorithm used to produce the Hash Result. SHA-1 = "01" hex.
Public Key Exponent Length	1 b	Length of CA Public Key Exponent (Number of bytes)
Public Key Exponent (*e*)	var b	Exponent of CA Public Key
Hash Result	20 b	Hash value from unsigned output extension

Self Signed CA Public Key Certificate

(Self-Signed Certificate)

Appendix M – Self Signed CA Public Key Certificate Format (Mastercard)

Field Name	Length & Format	Description		Hashed
ID of Certificate	5 b	The "Registered Application Provider Identifier" (RID)	NOT SIGNED	No
Public Key Index	1 b	Public Key Index, which uniquely identifies a Public Key.	NOT SIGNED	No
Public Key Algorithm Indicator	1 b	Indicates the algorithm to be used with the Public Key. RSA = "01" hex	NOT SIGNED	No
Public Key Length	1 b	Length of Public Key Modulus (equal to N_{CA})	NOT SIGNED	No
Public Key Exponent Length	1 b	Length of Public Key Exponent	NOT SIGNED	No
Leftmost Digits of Public Key	N_{CA}-37 b	N_{CA}-37 most significant bytes of the Public Key Modulus	NOT SIGNED	No
Public Key Remainder	37 b	37 least significant bytes of the Public Key Modulus	NOT SIGNED	Yes
Public Key Exponent	var b	Public Key Exponent	NOT SIGNED	Yes

Self Signed CA Public Key Certificate

(Clear Data)

Field Name	Length & Format	Description	Hashed
Recovered Data Header	1 b	Hex value '6A'	No
Certificate Format	1 b	Hex value '10'	Yes
ID of Certificate	5 b	The "Registered Application Provider Identifier" (RID)	Yes
Certificate Expiration Date	2 b	MMYY after which this certificate is invalid. See Note 1.	Yes
Certificate Serial Number	3 b	Inserted by scheme provider.	Yes
Hash Algorithm Indicator	1 b	Identifies the hash algorithm used to produce the Hash Result. SHA-1 = "01" hex.	Yes
Public Key Algorithm Indicator	1 b	Identifies the digital signature algorithm to be used with the Public Key. RSA = "01" hex	Yes
Public Key Length	1 b	Length of the Public Key Modulus in bytes (N_{CA})	Yes
Public Key Exponent Length	1 b	Length of Issuer Public Key Exponent in bytes (from 01 to 04)	Yes
Leftmost Digits of Public Key	N_{CA}-37 b	Leftmost N_{CA}-37 bytes of the Public Key Modulus	Yes
Hash Result	20 b	Hash of Public Key and its associated information.	No
Recovered Data Trailer	1 b	Hex value "BC"	No

Self Signed CA Public Key Certificate

(Self-Signed Certificate)

Appendix N - Self Signed CA Public Key Certificate Format (American Express)

Field Name	Length & Format	Description		Hashed
Header	1 b	Fixed value '20'	NOT SIGNED	No
Service Identifier (SI)	4 b	American Express Product Identifier Fixed value '00 00 00 00'	NOT SIGNED	No
Length of Public Key Modulus (N_{CA})	2 b	Number of bytes in Public Key Modulus (i.e., '80' = 1024 bits, '90' = 1152 bits, 'B0' = 1408 bits, 'F8' = 1984 bits) Length is right justified and padded with '00'	NOT SIGNED	No
Public Key Algorithm Identifier	1 b	Cryptographic algorithm to be used with the Public Key ('01' = RSA)	NOT SIGNED	No
Public Key Exponent Length	1 b	Number of bytes in Public Key Exponent (e.g., '01')	NOT SIGNED	No
Registered Application Provider Identifier (RID)	5 b	American Express Identifier = 'A000000025'	NOT SIGNED	Yes
Public Key Index	1 b	CA Key Pair Index	NOT SIGNED	Yes
Public Key Modulus (N_{CA})	var b	Public Key Modulus	NOT SIGNED	Yes
Public Key Exponent (e_{CA})	var b	Public Key Exponent (e_{CA} = 1 to N_{CA} / 4) (e.g., '03' Hexadecimal)	NOT SIGNED	Yes
Hash Result	20 b	Hash of the fields indicated above	NOT SIGNED	No

Self Signed CA Public Key Certificate

(Clear Data - Unsigned Output Extension)

Field Name	Length & Format	Description
Header	1 b	Fixed value '21'
Service Identifier (SI)	4 b	American Express Product Identifier Fixed value '00 00 00 00'
Registered Application Provider Identifier (RID)	5 b	American Express Identifier = 'A000000025'
Public Key Index	1 b	CA Key Pair Index
Certificate Expiration Date	2 b	MMYY after which the certificate is invalid (CA private key expiration date). See Note 1.
Public Key Algorithm Indicator	1 b	Cryptographic algorithm to be used with the Public Key ('01' - RSA)
Most Significant Part of Public Key Modulus (N_{CA})	var b	N_{CA} - [36+e_{CA}] bytes of the Modulus data. (36 = number of bytes in fixed fields)
Hash Algorithm Indicator	1 b	Cryptographic algorithm used to generate the Hash ('01' - SHA-1)
Public Key Exponent Length	1 b	Number of bytes in Public Key Exponent (e.g., '01')
Public Key Exponent (e_{CA})	var b	Public Key Exponent (e_{CA} = 1 to N_{CA} / 4) (e.g., '03' Hexadecimal)
Hash Result	20 b	Hash value from unsigned output extension

Self Signed CA Public Key Certificate

(Self-Signed Certificate)

Appendix O – DC_SUK block template

The DC_SUK block has JSON format. The 'IY' command accepts as input a JSON template having the form:

```
{
   "serviceData": {
      "DC_SUK_CONTENT": {
         "ATC": "*",
         "IDN": "*",
         "RFU": "*",
         "SK_CL_MD": "*",
         "SK_RP_MD": "*",
         "SUKInfo": "*",
         "SUK_CL_UMD": "*",
         "SUK_RP_UMD": "*",
         "hash": "*"
      },
      "DC_SUK_ID": "5413339000001513FFFF00150384090328000l150326"
   },
   "serviceID": "PROVISIONSUK",
   "serviceRequestID": "1427295513175"
}
```

The 'IY' command populates the JSON template replacing the wildcard '*' characters with the actual plain text key and data values for example:

```
{
   "serviceData": {
      "DC_SUK_CONTENT": {
         "ATC": "0001",
         "IDN": "DA98438667C8FDC2",
         "RFU": "00",
         "SK_CL_MD": "225BC8E86ED81A00F9CF9C74A6653BD5",
         "SK_RP_MD": "E8E486F384C8F1F8D5ED020E035391D8",
         "SUKInfo": "38",
         "SUK_CL_UMD": "FC24AF40DA0D2E8F5A83D7933CF521B6",
         "SUK_RP_UMD": "4F982FBF2186B4A7F82C45C25C0E216D",
         "hash": "01B05522B12D0BB1A61215F762D4005686CC5769"
      },
      "DC_SUK_ID":
"5413339000001513FFFF00150384090328000l150326A25EE16119ED291D2EE9651B"
   },
   "serviceID": "PROVISIONSUK",
   "serviceRequestID": "1427295513175"
}
```

Americas
Arboretum Plaza II, 9442 Capital of Texas Highway North,
Suite 100, Austin, TX 78759 USA
Tel: +1 888 343 5773 or +1 512 257 3900
Fax: +1 954 888 6211
E-mail: CPL_Sales_AMS_TG@thalesgroup.com

Asia Pacific
Unit 904-906A, Core D
Cyberport 3, 100 Cyberport Road
Pokfulam, Hong Kong | Tel: +852 3157 7111
Fax: +852 2815 8141 | E-mail: apacsales.cpl@thalesgroup.com

Europe, Middle East, Africa
350 Longwater Ave, Green Park,
Reading, Berkshire, UK RG2 6GF
Tel: +44 (0)1844 201800 | Fax: +44 (0)1844 208550
E-mail: CPL_Sales_EMEA_TG@thalesgroup.com

> cpl.thalesgroup.com <