THALES

cpl.thalesgroup.com

payShield 9000 v3.5

Host Command Examples

1270A646-038

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
1270A646-038	Issue 38	payShield 9000 v3.5	July 2120

Introduction

This document is designed to be used in conjunction with the *payShield 9000 Host Command Reference Manual*, the *payShield 9000 Host Programmer's Manual*, and the document *Applications Using the payShield 9000*. It will also be helpful in assisting developers generally when they are constructing host commands to be sent from their applications to a payShield 9000.

Chapter 2 provides examples of commonly used payShield 9000 host commands including sample parameters and data: the commands are presented in alphabetical order of their 2-character command IDs. It also provides the responses and output data that would be expected to the command. It is assumed that the Test LMKs provided with the payShield 9000 are being used. In Chapter 1 can be found two versions of Python code which can be used to send the example commands in Chapter 2 (or any other commands) to a payShield 9000 and to receive responses.

Chapter 1 - Sample Code

The unsupported sample Python code provided below lets software developers send individual commands to a payShield 9000 and receive responses back.

The code below runs the host command *HEADNC* (i.e. a 4-character header followed by the command code *NC*). The text *HEADNC* can be replaced by any desired host command. The example command text provided in Chapter 2 can be cut & pasted into the code below. Note that:

- Responses may be different from the examples in this document depending on security settings and the LMK being used. Most of the example commands use the Test LMKs delivered with the payShield 9000.
- This code is designed for use with Python v2.7. It may not run correctly if used with other versions of Python. The Python runtime environment can be downloaded free of charge from www.python.org .
- Copy the code within the text box below and save it as a file with a *.py* extension (for example *codeSample1.py*)
- After installing Python, the code can then be run by executing the following from the command line:

  ```
  python codeSample1.py
  ```

Code Sample 1

This code is very simple, and is suggested to be used for initial testing to get connections established with the HSM. The main limitation of this code is that it supports only Alphanumeric data in commands and responses, and cannot be used to send/receive binary data.

```
#!/usr/bin/python

import socket
from struct import *

# Change HSM IP and Port below to match deployment
TCP_IP = '192.168.0.60'
TCP_PORT = 1500
BUFFER_SIZE=1024

# paste command example within inverted commas below:
#e.g.:
COMMAND = 'HEADNC'

# 1st two bytes must be command length
SIZE=pack('>h',len(COMMAND))

# join everything together
MESSAGE = SIZE + COMMAND

# Create socket and connect
hsmSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
hsmSocket.connect((TCP_IP, TCP_PORT))

# send MESSAGE
hsmSocket.send(MESSAGE)

# receive
data = hsmSocket.recv(BUFFER_SIZE)

# close socket
hsmSocket.close()

print "sent data (ASCII)    : ", MESSAGE
print "sent data (HEX)      : ", MESSAGE.encode('hex')
print "received data (ASCII): ", data
print "received data (HEX)  : ", data.encode('hex')
```

Code Sample 2

This code supports the input and output of binary data, with the binary data being represented as hexadecimal characters and enclosed in angle brackets. For example, to enter the binary data *1001011001011110* the command input would include *<965E>* .

```
#!/usr/bin/python

import socket, binascii, string
from struct import *

# Change HSM IP and Port below to match deployment
TCP_IP = "192.168.0.60"
TCP_PORT = 1500

# paste command example within inverted commas below:
#e.g.:
COMMAND = 'HEADNC'

def testPrintable(str):
    return all(c in string.printable for c in str)

def buildCommand(command):
  #convert hex supplied data into binary
  hCommand = ''
  i = 0
  while True:
    if (command[i:i+1] == '<'):
      i = i + 1
      while True:
        hCommand = hCommand + binascii.a2b_hex(command[i:i+2])
        i = i + 2
        if (command[i:i+1] == '>'):
          i = i + 1
          break
    else:
      hCommand = hCommand + command[i]
      i = i + 1

    if (i == len(command)):
      break

  return hCommand

def main():
  global TCP_IP
  global TCP_PORT
  global COMMAND

  # Create socket and connect
  connection = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
  connection.connect((TCP_IP, TCP_PORT))

  BUFFER_SIZE = 1024
```

Continued on next page …

… continued from previous page.

```
  # Convert hex to binary
  COMMAND = buildCommand(COMMAND)

  # calculate the size and format it correctly
  SIZE=pack('>h',len(COMMAND))

  # join everything together
  MESSAGE = SIZE + COMMAND

  # send MESSAGE
  connection.send(MESSAGE)

  # receive data
  data = connection.recv(BUFFER_SIZE)

  # don't print ascii if msg or resp contain non printable chars
  if (testPrintable(MESSAGE[2:])):
    print "sent data (ASCII)    :", MESSAGE[2:]
  print "sent data (HEX)      :", MESSAGE.encode('hex')

  if (testPrintable(data[2:])):
    print "received data (ASCII):", data[2:]
  print "received data (HEX)  :", data.encode('hex')

  connection.close()

if __name__ == "__main__":
  main()
```

Chapter 2 – Example Host Commands

Notation

In the "Format" column in the following examples the following abbreviations are used:

- n – represents a variable number of characters or bytes.
- A – represents Alphanumeric (ASCII or EBCDIC) characters.
- H – represents hexadecimal digits ("0..9" and "A..F")
- B – represents Bytes of binary data
- C – represents Control characters

In the following examples of input/output data, <…> indicates binary data represented as Hexadecimal characters to allow the data to be shown as printable characters in this document. In reality the data would be sent/received as bytes of binary data. The "<" and ">" are not part of the sent/received data.

A0 Command - Generate a TPK

The A0 command is used to generate DES or TDES keys. (The EI command is used to generate RSA key pairs – see later.) The key will be encrypted under the appropriate LMK variant or in an LMK-encrypted key block with appropriate parameters. Optionally the key can also be encrypted under a TMK or ZMK for export to other systems.

In the example below, we are generating a double-length TDES TPK, which is also to be exported under a double-length TDES ZMK using X9.17 methods. (*Note: for this to be allowed, the security settings must allow export using X9.17 and to allow export in non-trusted format.*)

The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
HDR1A01002U;1U3E8C28EBF7D3799DC7147E3441F8E452X
```

Field	Format	Value	Description
1	4 A	HDR1	Message header (as defined by user).
2	2 A	A0	Command Code. (Note: the second character is a numeric zero.)
3	1 H	1	Mode. 0 = Generate key and encrypt under a TMK
4	3 H	002	Key Type. 002 = TPK
5	1 A	U	Key Scheme - for encrypting the output key under the LMK. U = Double-length TDES key scheme
6	1 A	;	Delimiter

Field	Format	Value	Description
7	1 N	1	TMK/ZMK Flag 1 = Key is to be exported under a TMK.
8	1 A + 32 H	U3E8C28EBF7D3799DC71 47E3441F8E452	The TMK (encrypted under LMK pair 14-15 variant 0 or 36-37 variant 8, depending on whether PCI HSM compliance has been enforced) that the exported TPK is to be encrypted under.
9	1 A	X	Key Scheme for exporting the TPK encrypted under the TMK. X = Encryption of a double length key using X9.17 methods

Response

```
HDR1A100UF4395AE377FA12297721AB182E96C552XE2C6964890CEF33393BA
FAD61D26D11ADD60A7
```

Field	Format	Value	Description
1	4 A	HDR1	Message header (as defined by user).
2	2 A	A1	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	1 A+ 32 H	UF4395AE377FA1229772 1AB182E96C552	The generated TPK, encrypted under LMK pair 14-15 variant 0 or 36-37 variant 7, depending on whether PCI HSM compliance has been enforced. U = Double-length TDES key scheme
5	1 A+ 32 H	XE2C6964890CEF33393B AFAD61D26D11A	The generated TPK encrypted under the TMK
6	6 H	DD60A7	Key Check Value

A0 Command - Generate an IKEY/IPEK

In this example for A0, we are deriving a double-length TDES DUKPT Initial Key (a.k.a. Initial PIN Encryption Key) from a double-length TDES BDK (Type 2). The key does not need to be exported.

The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
HDR2A0A302U02U000595797DB29310FDF463ABB8396F451016E3658200007
```

Field	Format	Value	Description
1	4 A	HDR2	Message header (as defined by user).
2	2 A	A0	Command Code. (Note: the second character is a numeric zero.)
3	1 H	A	Mode. A = Derive key (from a BDK)
4	3 H	302	Key Type. 302 = IKEY/IPEK, encrypted under Variant LMK pair 14-15.
5	1 A	U	Key Scheme - for encrypting the output key under the LMK. U = Double-length TDES key scheme
6	1 A	0	Derive Key Mode 0 = Derive IKEY/IPEK from BDK
7	1 H	2	DUKPT Master Key Type. 2 = Type 2 BDK
8	1 A + 32 H	U000595797DB29310FDF463ABB8396F45	The Type 2 BDK encrypted under LMK pair 28-29 variant 6
9	15 H	1016E3658200007	KSN (Key Serial Number), consisting of a Key Set Identifier and Device ID for deriving the Initial Key. (NOTE: there is no Transaction Counter, as there would be with a KSN sent in transaction data from a terminal.)

Response

```
HDR2A100U4D374057BB0CB56246B0417CE2319C8175CA90
```

Field	Format	Value	Description
1	4 A	HDR2	Message header (as defined by user).
2	2 A	A1	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	1 A + 32 H	U4D374057BB0CB56246B0417CE2319C81	The generated IKEY/IPEK, encrypted under Variant LMK pair 14-15, variant 3. U = Double-length TDES key scheme
5	6 H	75CA90	Key Check Value (KCV).

CA Command - Translate a PIN from TPK to ZPK Encryption

The CA command is used to translate a PIN Block received from an ATM or POS terminal (and encrypted under a TPK) into PIN Block encrypted under a ZPK for onward transmission into the payments network. Different formats can be used for the received and translated PIN Blocks, but note that some standards (e.g. PCI HSM) limit which PIN Block format translations are permitted.

The CA command is used as the benchmark for specifying the payShield 9000. For example, the Model X payShield 9000, rated at 1,500 tps, can process 1,500 CA transactions (commands) per second.

In the example here, both the TPK and ZPK are double-length TDES keys, and Thales Format 01 (= ISO/ANSI Format 0) is used for both PIN Blocks.

The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
HDR7CAU1750CDFB0757D3B3994430636DBB281BUA787EF2F6595A8158EECE4
2B1170228912BFB87728013AD7610101234567812345
```

Field	Format	Value	Description
1	4 A	HDR7	Message header (as defined by user).
2	2 A	CA	Command Code.
3	1 A + 32 H	U1750CDFB0757D3B3994430636DBB281B	The TPK (encrypted under LMK pair 14-15 variant 0 or 36-37 variant 7, depending on whether PCI HSM compliance has been enforced) under which the PIN block is currently encrypted. U = Double-length TDES key scheme
4	1 A + 32 H	UA787EF2F6595A8158EECE42B11702289	The ZPK (encrypted under LMK pair 06-07 variant 0) under which the PIN block is to be encrypted. U = Double-length TDES key scheme
5	2 N	12	Maximum PIN Length -must be between 04 and 12.
6	16 H	BFB87728013AD761	Source PIN Block, encrypted under the TPK.
7	2 N	01	Source PIN Block format code. 01 = ISO 9564-1 & ANSI X9.8 format 0.
8	2 N	01	Output PIN Block format code. 01 = ISO 9564-1 & ANSI X9.8 format 0.
9	12 N	234567812345	Account Number. For output PIN Block format code = 01, the 12 right-most digits of the PAN excluding the check digit.

Response

```
HDR7CB000447125015E32C3FA301
```

Field	Format	Value	Description
1	4 A	HDR7	Message header (as defined by user).
2	2 A	CB	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	2 N	04	Length of the returned PIN.
5	16 H	47125015E32C3FA3	Output PIN Block
5	2 N	01	Output PIN Block format code - as provided in the command. 01 = ISO 9564-1 & ANSI X9.8 format 0.

CA Command - Translate a PIN from TPK to ZPK Encryption (using keys and PIN Block held in user storage)

In this example, rather than the encrypted TPK, ZPK, and PIN Block being included in the command data, the command includes indexes to these items stored within the HSM's user storage area.

The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
HDR7CAUK000UK00212K0040101234567812345
```

Field	Format	Value	Description
1	4 A	HDR7	Message header (as defined by user).
2	2 A	CA	Command Code.
3	2 A + 3 H	UK000	Index to the TPK (encrypted under LMK pair 14-15 variant 0 or 36-37 variant 7, depending on whether PCI HSM compliance has been enforced) held in user storage. First character is used to define the key scheme (e.g. "U" = Double-length TDES key scheme) or is not present for single-length DES. The second character of "K" indicates that a key reference is being used. The final 3 hex characters, "000", are the index to where the key is stored in user storage. If the key is stored in multiple indices (e.g. the block size has been set to "Single" but a double-length key has been stored) then this index points to the first of the blocks where the key is stored.
4	2 A + 3 H	UK002	Index to the ZPK (encrypted under LMK pair 06-07 variant 0) under which the PIN block is to be encrypted, held in user storage. "U" = Double-length TDES key scheme. The second character of "K" indicates that a key reference is being used. The final 3 hex characters, "002", are the index to where the key is stored in user storage. If the key is stored in multiple indices (e.g. the block size has been set to "Single" but a double-length key has been stored) then this index points to the first of the blocks where the key is stored.
5	2 N	12	Maximum PIN Length -must be between 04 and 12.
6	16 H	K004	Index to the Source PIN Block encrypted under the TPK, held in user storage. "K" indicates that a key reference is being used. The final 3 hex characters, "004", are the index to where the PIN Block is stored in user storage.
7	2 N	01	Source PIN Block format code. 01 = ISO 9564-1 & ANSI X9.8 format 0.
8	2 N	01	Output PIN Block format code. 01 = ISO 9564-1 & ANSI X9.8 format 0.
9	12 N	234567812345	Account Number. For output PIN Block format code = 01, the 12 right-most digits of the PAN excluding the check digit.

Response

The response will be the same as in the preceding example.

CA Command - Translate a PIN from TPK to ZPK Encryption (Key Block LMK)

In this example of the use of the CA command, we are using a key block LMK rather than a variant LMK. Both the TPK and the ZPK are included in the command in the format of a Thales Key Block, which contains metadata about the key as well as the encrypted key and a MAC.

The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
HDR8CAS0005671DN00N00026C2A641FEA15E950310D601D7E274B4C7403213
9S0005672DN00N00022BA44D32EE85E5BCEF74AA3C376D8C18EF702A70123C
8FFAD1C74C12AB0102123456789012
```

Field	Format	Value	Description
1	4 A	HDR8	Message header (as defined by user).
2	2 A	CA	Command Code.
3	1 A + 56 H	S0005671DN00N00026C2A641FEA15E950310D601D7E274B4C74032139	The TPK (encrypted under a Key Block LMK) under which the PIN block is currently encrypted. S = Key Block key scheme 0 = Version 0056 = total length of key block 71 = key usage (TPK) D = Algorithm (DES) N = Mode of Use (No special restrictions) 00 = Key Version Number N = Exportability (No Export permitted) 00 =number of optional header blocks 02 = LMK Identifier 6C2…B4C = Encrypted TPK 74032139 = MAC
4	1 A + 56 H	S0005672DN00N00022BA44D32EE85E5BCEF74AA3C376D8C18EF702A70	The ZPK (encrypted under a Key Block LMK) under which the PIN block is to be encrypted. S = Key Block key scheme 0 = Version 0056 = total length of key block 72 = key usage (ZPK) D = Algorithm (DES) N = Mode of Use (No special restrictions) 00 = Key Version Number N = Exportability (No Export permitted) 00 =number of optional header blocks 02 = LMK Identifier 2BA…C18 = Encrypted ZPK EF702A70 = MAC
5	2 N	12	Maximum PIN Length - must be between 04 and 12.
6	16 H	3C8FFAD1C74C12AB	Source PIN Block, encrypted under the TPK.
7	2 N	01	Source PIN Block format code. 01 = ISO 9564-1 & ANSI X9.8 format 0.
8	2 N	02	Output PIN Block format code. 02 = Docutel ATM format.
9	12 N	123456789012	Account Number.

Response

```
HDR7CB0004B9392CE3ABBA5BE702
```

Field	Format	Value	Description
1	4 A	`HDR7`	Message header (as defined by user).
2	2 A	`CB`	Response Code.
3	2 A	`00`	Error Code. `00` = No error.
4	2 N	`04`	Length of the returned PIN.
5	16 H	`B9392CE3ABBA5BE7`	Output PIN Block
5	2 N	`02`	Output PIN Block format code - as provided in the command. `02` = Docutel ATM format..

DE Command - Generate an IBM PIN Offset

The DE command allows for customer-selected PINs where the "natural" PIN has been generated using the IBM 3624 method (e.g. by using the EE host command), by calculating the offset between the natural and selected PINs.

The command is required because the card issuer does not know the natural PIN – they have access only to the natural PIN encrypted under the LMK. Note that the LMK-encrypted PIN uses an algorithm which involves the account number, and therefore the LMK-encrypted PIN for two different cardholders will be different even if they have the same cleartext PIN.

In this example we are using a single-DES PVK. The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
HDRCDE9A7EC8F365E1C0B01582604423046815398DD0BDFA69B41010C11223
3N44556
```

Field	Format	Value	Description
1	4 A	HDRC	Message header (as defined by user).
2	2 A	DE	Command Code.
3	16 H	9A7EC8F365E1C0B0	PVK that is used to generate the offset. The PVK is encrypted under LMK pair 14-15 variant 0.
4	10 N	1582604423	LMK-encrypted PIN. The length of this field is determined by the "PIN Length" security setting + 1.
5	2 N	04	Minimum PIN Length
6	12 N	423046815398	The 12 right-most digits of the PAN, excluding the check digit.
7	16 H	DD0BDFA69B41010C	Encrypted Decimalisation Table.
8	12 A	112233N44556	User-defined data. The "N" indicates where the last 5 digits of the PAN should be inserted.

Response

```
HDRCDF000835FFFFFFFF
```

Field	Format	Value	Description
1	4 A	HDRC	Message header (as defined by user).
2	2 A	DF	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	12 H	0835FFFFFFFF	The resulting Offset, F-padded to the left.

DE Command - Generate an IBM PIN Offset (using a Decimalisation Table held in User Storage)

The DE command allows for customer-selected PINs where the "natural" PIN has been generated using the IBM 3624 method (e.g. by using the EE host command), by calculating the offset between the natural and selected PINs.

The command is required because the card issuer does not know the natural PIN – they have access only to the natural PIN encrypted under the LMK. Note that the LMK-encrypted PIN uses an algorithm which involves the account number, and therefore the LMK-encrypted PIN for two different cardholders will be different even if they have the same cleartext PIN.

In this example we are using a single-DES PVK. The Decimalisation Table is held in the User Storaget area. The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
HDRCDE9A7EC8F365E1C0B01582604423046815398DD0BDFA69B41010C11223
3N44556
```

Field	Length	Value	Description
1	4 A	HDRC	Message header (as defined by user).
2	2 A	DE	Command Code.
3	16 H	9A7EC8F365E1C0B0	PVK that is used to generate the offset. The PVK is encrypted under LMK pair 14-15 variant 0.
4	10 N	1582604423	LMK-encrypted PIN. The length of this field is determined by the "PIN Length" security setting + 1.
5	2 N	04	Minimum PIN Length
6	12 N	423046815398	The 12 right-most digits of the PAN, excluding the check digit.
7	16 H	K123	Reference to the Encrypted Decimalisation Table, held in user storage at index 123. (Note: first character is the Index Flag and always has a value of 'K', irrespective of the value of the Index Flag used when the data was read into user storage using the LA command.)
8	12 A	112233N44556	User-defined data. The "N" indicates where the last 5 digits of the PAN should be inserted.

Response

```
HDRCDF000835FFFFFFFF
```

Field	Length	Value	Description
1	4 A	HDRC	Message header (as defined by user).
2	2 A	DF	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	12 H	0835FFFFFFFF	The resulting Offset, F-padded to the left.

DG Command - Generate a PVV of an LMK-encrypted PIN

The DG command is used to calculate the PVV from the PIN – e.g. after the cardholder has selected a new PIN. The PIN is provided in the command as an LMK-encrypted PIN, which uses an algorithm involving the account number, and therefore the LMK-encrypted PIN for two different cardholders will be different even if they have the same cleartext PIN. The command can optionally check the PIN against a list of weak or excluded PINs.

In the example below, the PVK pair are each single-length DES keys. Two excluded PINs are provided in the command, and LMK with ID=02 is to be used.

The user has specified a header (with the length set to 4 characters in the host port settings) and a trailer.

Command

```
HDRDDG445ACC722DB0B8854A1DAAD18F1B59A615826423046815398 3*02041
2341111%03<19>TRLRD
```

Field	Format	Value	Description
1	4 A	HDRD	Message header (as defined by user).
2	2 A	DG	Command Code.
3	32 H	445ACC722DB0B885 4A1DAAD18F1B59A6	PVK Key Pair, each encrypted under LMK Pair 14-15 variant 0.
4	5 N	15826	LMK-encrypted PIN. The length of this field is determined by the "PIN Length" security setting + 1.
5	12 N	423046815398	The 12 right-most digits of the PAN, excluding the check digit.
6	1 N	3	PVKI. Valid range is "0" to "6".
7	1 A	*	Delimiter, indicating that an excluded PIN table follows.
8	2 N	02	Number of entries in the excluded PIN table.
9	2 N	04	Length of each encrypted PIN (must be in range 04-12).
10	8 N	12341111	Excluded PIN table – i.e. PINs "1234" and "1111" are not allowed.
11	1 A	%	Delimiter to indicate that an LMK ID follows
12	2N	03	LMK 03 must be used for this command.
13	1 C	<19>	Delimiter to indicate that a trailer follows.
14	4 A	TRLRD	Trailer, as defined by user.

Response

```
HDRDDH005809<19>TRLD
```

Field	Format	Value	Description
1	4 A	HDRD	Message header (as defined by user).
2	2 A	DH	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	4 N	5809	The output PVV
5	1 C	<19>	Delimiter to indicate that a trailer follows.
6	4 A	TRLRD	Trailer, as provided in command.

EE Command – Derive a PIN using the IBM Offset Method

The EE command derives a "natural" PIN based on the account number, using the IBM 3642 method.

A single-length DES PVK is being used to derive the PIN.

The user has specified a header (with the length set to 4 characters in the host port settings) and a trailer.

Command

```
HDRDEE9A7EC8F365E1C0B02468FFFFFFFF04423046815398DD0BDFA69B4101
0C112233N44556<19>TRLRD
```

Field	Format	Value	Description
1	4 A	HDRD	Message header (as defined by user).
2	2 A	EE	Command Code.
3	16 H	9A7EC8F365E1C0B0	PVK Key Pair used to derive the PIN, each encrypted under LMK Pair 14-15 variant 0.
4	12 H	2468FFFFFFFF	Offset, padded to the right with hexadecimal digit F.
5	2 N	04	Minimum PIN length
6	12 N	423046815398	Account number – 12 rightmost digits excluding the check digit.
7	16 H	DD0BDFA69B41010C	Decimalization table.
8	12 A	112233N44556	User defined PIN Validation Data. The "N" character indicates where the last 5 digits of the account number should be inserted.
9	1 C	<19>	Delimiter to indicate that a trailer follows.
10	4 A	TRLRD	Trailer, as provided in command.

Response

```
HDRDEF0088949<19>TRLRD
```

Field	Format	Value	Description
1	4 A	HDRD	Message header (as defined by user).
2	2 A	EF	Response Code.
3	5 N	88949	The derived PIN encrypted under the LMK. The length of this field is determined by the "PIN Length" security setting + 1.
4	1 C	<19>	Delimiter to indicate that a trailer follows.
5	4 A	TRLRD	Trailer, as provided in command.

EI Command - Generate RSA Key Pair

The EI command is used to generate RSA key pairs. (A0 command is used to generate DES and TDES keys – see the examples earlier in this section).

In this example we are generating a key pair for use in key management with modulus length of 1,024 bits. The user has specified a header (with the length set to 4 characters in the host port settings) and a trailer.

Command

```
HDR9 EI1102401<19>4RH486ET4U8J4R5874T8
```

Field	Format	Value	Description
1	4 A	HDR9	Message header (as defined by user).
2	2 A	EI	Command Code.
3	1 N	1	Key type indicator. 0 = key management only.
4	4 N	1024	Modulus length in bits.
5	2 N	01	Encoding rules for the public key. 01 = DER encoding for ASN.1 Public Key (INTEGER uses unsigned representation).
6	1 C	<19>	Delimiter, indicating that there is a trailer for the message.
7	20 A	4RH486ET4U8J4R5874T8	Message trailer, defined by user. (Max length is 32 chars.)

Response

```
HDR9EJ00<308188028180ADAF406F3901EC0F2583BB1945C49D7B3924A4AFC
7833FFC4C03BF13134DA8E3F28DE306E40D83B7975AC45D128D24EFF005750
5F1FEC90A7EA4CBB483A884168E85E60939A72F74945D3D3889BD41B564736
E8FC90D26B1E97FBFF03A7B584E2AFEAE39D19F00798782FCA09BEF18A3F30
27900DF5A1E0ED50384AED1C52FAF0203010001>0344<AC5DC5A50CA3D9293
629994FF8452E767B79090C9F6B86C60149473B5EAFDF109828250DF286542
120A4EFB678694B244A8C18F5889109F6EBB226200559212730C13609274A5
816A6074DC7AAE39B707EB5B3F2AA7D2EAAA10A6D3A6EFC784A7749F0800E7
84F0E6E03921317426D4DA8522E50E5063DA1F2AEC33018E7A6BC07AA2914D
42E351BB88A2FAF3D581D6D16D800D6FFAF51FCABF1F95A9429AF94E564C54
9B0FFF261C8310EE165DCBA3E7CEF83421D3FAEB40FD7A1BBDD620EEA44B42
33D15A7F9FBBB8BAB8D2016601A43110F7EF5CACAC282E82D5D580171EEB05
15422B71CC2F746C427D48B9F3FAF0D2B8F7186F0E5D63A9946E97F08BB974
43FD700D641169FD975792EF7FA79E78CDFB85973EE14F7F1234478F947ACB
472B51130FD2C4959B7835D49CF2FA014F17CC913CD8A4D2D92ABF6A3BE6FB
E0E12026077C2FDFCDA1D825402E0399AC47527F7E881BA68F1><19>4RH486
ET4U8J4R5874T8
```

Field	Format	Value	Description
1	4 A	HDR9	Message header (as defined by user).
2	2 A	EJ	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	392 bits	<3081…0001>	Public key, encoded as required by field 5 of the command.
5	4 N	0344	Length of Private key, in bytes.
6	120 B	<AC5D…68F1>	Private key, encrypted under the LMK.
7	1 C	<19>	Delimiter, indicating that there is a trailer for the message.
8	20 A	4RH486ET4U8J4R5874T8	Message trailer as provided in the command.

EO Command - Import a Public Key

This command allows a public RSA key to be imported by generating a MAC on it (variant LMK) or creating a key block (key block LMK). This might be used to protect a CA public key.

In the following example, we are importing a public key. The user has specified a header (with the length set to 4 characters in the host port settings) and a trailer.

Command

```
HDRAEO01<30470240AE213EBDFE272616058114CA5D6E8DDD2F976EF0C6D6F
6ADE0204E916CE0B5C704A1A9F1F631F578D7B87D1981951BB99A572400AC4
3BB3AF8C2A4068A005225020301000l><41555448454E2D44415441><19>GF
6W54G65WR456GAER54GE6R45G6E4AR
```

Field	Format	Value	Description
1	4 A	HDRA	Message header (as defined by user).
2	2 A	EO	Command Code. (NOTE: the 2nd character is an alphabetic "O" for Oscar.)
3	2 N	01	Encoding rules for the public key. 01 = DER encoding for ASN.1 Public Key (INTEGER uses unsigned representation).
4	584 bits	<3047…0001>	Public key, encoded as required by field 3.
5	88 bits	<4155…5441>	Additional data for the MAC calculation.
6	1 C	<19>	Delimiter, indicating that there is a trailer for the message.
7	32 A	GF6W54G65WR456GAER54GE 6R45G6E4AR	Message trailer, defined by user. (Max length is 32 chars.)

Response

```
HDRAEP00<9C350891><30470240AE213EBDFE272616058114CA5D6E8DDD2F9
76EF0C6D6F6ADE0204E916CE0B5C704A1A9F1F631F578D7B87D1981951BB99
A572400AC43BB3AF8C2A4068A0052250203010001><19>GF6W54G65WR456GA
ER54GE6R45G6E4AR
```

Field	Format	Value	Description
1	4A	HDR9	Message header (as defined by user).
2	2A	EP	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	4 Bytes	<9C350891>	MAC on the public key and additional data, calculated using LMK key pair 36-37.
5	584 bits	<3047…0001>	The imported Public Key, encoded as per field 3 of the command.
6	1 C	<19>	Delimiter, indicating that there is a trailer for the message.
7	32 A	GF6W54G65WR456GAER54GE 6R45G6E4AR	Message trailer, as provided in the command.

EW Command - Generate a signature (with variant LMK-encrypted RSA key provided in the command)

The EW command generates a signature over a message using an RSA private key encrypted under a variant LMK. In this example, the variant LMK-encrypted RSA private key is included in the command.

Command

```
0320EW0201010013<61626364656768696061626364>;990264<6783CD3B9B
468B622190C50C91D7559CB1654D2BBEB41F449CB202A910B5AB7B1716A2B1
3012403DD1333E264BE0ED677B0ADF8F5B849C3E9312887E095F3A97E6AA9A
7B33E23AE2E741801ADA0589173C22BC6355368B737BA6C78781D0308EDF4E
7A7AD833FF7061BE6B988C9399A13FDDDF66EAAEEC3744016ACEA6FCF49F2B
F69D45784F561324B0D4DC9EF65373E4566DEFC2B67F328780E38FEEA74519
F63BCE9712C82A8763DEEBFDFF24CA2CA98A7C1DC799A897C3AD0C25355A9F
0CD9D7262C84D2D1C5C4556E34887B6975AEC66E2A244EBAE3C582CFD91A87
47A735BCE60B905F6053691440C9BB2E658C27D77B2752674026A1676EB9EC
5FC6B158A4916C92864040>
```

Field	Format	Value	Description
1	4 A	0320	Message header (as defined by user).
2	2 A	EW	Command Code.
3	2 N	02	Hash identifier. (02 = MD5)
4	2 N	01	Signature identifier. (01 = RSA)
5	2 N	01	Pad mode identifier (01 = PKCS#1 v1.5)
6	4 N	0013	Data Length
7	13 B	<61626364656768696061626364>	Message to be signed
8	1 A	;	Delimiter
9	2 N	99	Private key flag. (99 = key identified below.)
10	4 N	0264	Private key length
11	264 B	<6783CD3B9B468B622190C50C91D7559CB1654D2BBEB … 77B2752674026A1676EB9EC5FC6B158A4916C92864040>	Private RSA key used to generate the signature

Response

```
0320EX000097<008226CE8A1FD195DF87975DF65A810182700EABC7C73DBC7
520440C2C48230BCB8749B843A0B22FC2C16163288A182710B4BFFFBC870AB
A67BBDD74F3DABB3A1FB1DB8BD69DE7D384C69167530DD706302509291AC60
239CEC0EA5774A372762A>
```

Field	Format	Value	Description
1	4 A	0320	Message header (as defined by user).
2	2 A	EX	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	4 N	0097	Length of signature
5	97 B	<008226CE8A1FD195DF879 75DF65A810182700EABC7C … 30DD706302509291AC6023 9CEC0EA5774A372762A>	Calculated signature

EW Command - Generate a signature (with variant LMK-encrypted RSA key held in user storage)

The EW command generates a signature over a message using an RSA private key encrypted under a variant LMK. In this example, the variant LMK-encrypted RSA private key is held in user storage at index 000, with the block size set to **V**ariable in security settings. *(NOTE: the ability to reference RSA keys in user storage is not available in payShield 9000 v2.2a or earlier.)*

Command

```
0310EW0101010009<616263646567686960>;990004K000
```

Field	Format	Value	Description
1	4 A	0310	Message header (as defined by user).
2	2 A	EW	Command Code.
3	2 N	01	Hash identifier. (01 = SHA-1)
4	2 N	01	Signature identifier. (01 = RSA)
5	2 N	01	Pad mode identifier (01 = PKCS#1 v1.5)
6	4 N	0009	Data Length
7	9 B	<616263646567686960>	Message to be signed
8	1 A	;	Delimiter
9	2 N	99	Private key flag. (99 = key identified below.)
10	4 N	0004	Private key length
11	'K' + 3 H	K000	Index to RSA private key in user storage ("K" = the index flag; "000" is index of key in user storage).

Response

```
0310EX000064<418AFB0364BCD82C2240EAABFDBBA0F9C0DD84319005D8686
BD77ADF8ACA59C8513B86D76444B757E30B081B2D05F2A888354EEEEE134A6
370FA528F34C82163>
```

Field	Format	Value	Description
1	4 A	0310	Message header (as defined by user).
2	2 A	EX	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	4 N	0064	Length of signature
5	64 B	<418AFB0364BCD82C2240E AABFDBBA0F9C0DD8431900 … B2D05F2A888354EEEEE134 A6370FA528F34C82163>	Calculated signature

EW Command - Generate a signature (with key block LMK-encrypted RSA key held in user storage)

The EW command generates a signature over a message using an RSA private key encrypted under a key block LMK. In this example, the key block LMK-encrypted RSA private key is held in user storage at index 022, with the block size set to **V**ariable in security settings. *(NOTE: the ability to reference RSA keys in user storage is not available in payShield 9000 v2.2a or earlier.)*

Command

```
0310EW0101010003<616263>;99FFFFSK022%02
```

Field	Format	Value	Description
1	4 A	0310	Message header (as defined by user).
2	2 A	EW	Command Code.
3	2 N	01	Hash identifier. (01 = SHA-1)
4	2 N	01	Signature identifier. (01 = RSA)
5	2 N	01	Pad mode identifier (01 = PKCS#1 v1.5)
6	4 N	0003	Data Length
7	3 B	<616263>	Message to be signed
8	1 A	;	Delimiter
9	2 N	99	Private key flag. (99 = key identified below.)
10	4 N	FFFF	Private key length. (Always set to "FFFF" for key block LMKs.)
11	'K' + 3 H	SK022	Index to RSA private key in user storage ("S" identifies key block LMK scheme; "K" = the index flag; "022" is index of key in user storage. (Note: the index flag always has a value of 'K', irrespective of the value of the Index Flag used when the data was read into user storage using the LA command.)).
12	1 A	%	Delimiter
13	2 N	02	LMK identifier.

Response

```
0310EX000128<02658202FF77FE249356E9A4B2B43771A87A1B59518B64955
A6AA4C56DA268FF65521BA4C9A594D136354DA2F83040988AD3115224A29CF
5CE2F8CF084EA1E4ED7C3BC5C59C023F0CB59B53F5ECE58A133E2682B2BC16
00749641EAC5BDE729FAB66715DF7566C5CAD80E443DC5519A11690FB5B5BB
C0BF9842832869284 6856>
```

Field	Format	Value	Description
1	4 A	0310	Message header (as defined by user).
2	2 A	EX	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	4 N	0128	Length of signature
5	128 B	<02658202FF77FE249356E 9A4B2B43771A87A1B59518 … 15DF7566C5CAD80E443DC5 519A11690FB5B5BBC0BF98 428328692846856>	Calculated signature

EY Command - Validate a signature (with variant LMK-encrypted RSA key provided in the command)

The EY command validates, using the RSA public key, a signature previously generated over a message using an RSA private key. In this example, the variant LMK-encrypted RSA private key is included in the command.

Command

```
0350EY0301010040<6F3291898D0488CB52478A84628596617CE7F92BB2272
4771072F0885F823C5210AEDA2CAE0A6848>;0003<616263>;<60973C58><3
02F0228B4FA9DC06844F408116C0B8EF7D1046593F73FB44F432E3732891C4
E75F5C0EA07D968D2FFB5E6250203010001><41555448454E2D44415441>
```

Field	Format	Value	Description
1	4 A	0350	Message header (as defined by user).
2	2 A	EY	Command Code.
3	2 N	03	Hash identifier. (03 = ISO 10118-2)
4	2 N	01	Signature identifier. (01 = RSA)
5	2 N	01	Pad mode identifier (01 = PKCS#1 v1.5)
6	4 N	0040	Signature Length
7	40 B	<6F3291898D0488CB52478A84628596617CE7F92BB22724771072F0885F823C5210AEDA2CAE0A6848>	Signature to be validated
8	1 A	;	Delimiter
9	4 N	0003	Length of data in message for which the signature is to be validated.
10	3B	<616263>	Message whose signature is to be validated.
11	1 A	;	Delimiter
12	4 B	<60973C58>	MAC on the Public Key and Authentication Data
13	49 B	<302F0228B4FA9DC06844F408116C0B8EF7D1046593F73FB44F432E3732891C4E75F5C0EA07D968D2FFB5E6250203010001>	RSA Public Key
14	11 B	<41555448454E2D44415441>	Optional Authentication Data.

Response

```
0350EZ00
```

Field	Format	Value	Description
1	4 A	0350	Message header (as defined by user).
2	2 A	EZ	Response Code.
3	2 A	00	Error Code. 00 = No error. Other codes would indicate the reason why the signature was not validated.

G0 Command - Translate a PIN block from BDK to ZPK Encryption

The G0 command performs a function analogous to the CA command described earlier, but is used for DUKPT terminals. It translates a PIN Block received from a DUKPT terminal into a PIN Block (encrypted under a ZPK) that can be passed into the payments network. The key used to encrypt the PIN is not provided to the payShield 9000: instead, it derives it from the BDK provided to it (which is identified to the host system in the KSN sent from the terminal) and the KSN itself. The command can also translate between different PIN Block formats.

In this example, the received PIN Block will be decrypted using a double-length TDES BDK and the transmitted PIN Block is encrypted using a double-length TDES ZPK. Both received and transmitted PIN Blocks use Thales format 01 (= ISO/ANSI Format 0).

The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
HDR4G0~U000595797DB29310FDF463ABB8396F45U16FAD09093F342EC35AE6
049B4F80834609FFFF01016E3658200007080C2A27DF4D6A4F010134567891
2345
```

Field	Format	Value	Description
1	4 A	HDR4	Message header (as defined by user).
2	2 A	G0	Command Code. (NOTE: the 2nd character is a zero.)
3	1 A	~	BDK Flag. ~ indicates that the supplied BDK is a Type 2 BDK.
4	1 A + 32 H	U000595797DB29310FDF463ABB8396F45	The Type 2 BDK (encrypted under LMK pair 28-29 variant 6) required to decrypt the PIN Block. U = Double-length TDES key scheme
5	1 A + 32 H	U16FAD09093F342EC35AE6049B4F80834	The ZPK (encrypted under LMK pair 06 07) to be used to re-encrypt the PIN Block. U = Double-length TDES key scheme
6	3 H	609	KSN Descriptor. 6 = BDK identifier length (in Hex chars.) 0 = Sub-key identifier length (in Hex chars.) 9 = Device identifier length (in Hex chars.)
7	20 H	FFFF01016E3658200007	KSN (Key Serial Number) FFFF02 = BDK identifier 016E36582 = Device identifier 00003 = Transaction Counter
8	16 H	080C2A27DF4D6A4F	Source PIN Block, encrypted using DUKPT
9	2 N	01	Source PIN Block format code. 01 = ISO 9564-1 & ANSI X9.8 format 0.
10	2 N	01	Output PIN Block format code. 01 = ISO 9564-1 & ANSI X9.8 format 0.
11	12 N	345678912345	Account Number. For output PIN Block format code = 01, the 12 right-most digits of the PAN excluding the check digit.

Response

```
HDR4G10004BE09A161870F9E8F01
```

Field	Format	Value	Description
1	4 A	HDR4	Message header (as defined by user).
2	2 A	G1	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	2 N	04	PIN Length
5	16 H	BE09A161870F9E8F	Output PIN Block, encrypted under the ZPK.
6	2 N	01	Output PIN Block format code - as in the command, field 10. 01 = ANSI X9.8 Format 0.

GM Command – Hash a block of data

The GM command calculates a hash over a block of data of up to 32,000 Bytes.

The user has specified a header (with the length set to 4 characters in the host port settings), and a trailer.

Command

```
4000GM0100032<000000000000000000000000000000000000000000000000
0000000000000000><19> = 2EF96
```

Field	Format	Value	Description
1	4 A	4000	Message header (as defined by user).
2	2 A	GM	Command Code.
3	2 N	01	Hash algorithm to be used. 01 =SHA-1.
4	5 N	00032	Length of message over which hash is to be calculated.
5	32 B	<000000000000000000000 0000000000000000000000 000000000000000000000>	Message to be hashed.
6	1 C	<19>	Delimiter, indicating that there is a trailer for the message.
7	8 A	= 2EF96	Message trailer, defined by user. (Max length is 32 chars.)

Response

```
4000GN00<DE8A847BFF8C343D69B853A215E6EE775EF2EF9619> = 2EF96
```

Field	Format	Value	Description
1	4 A	4000	Message header (as defined in the command).
2	2 A	GN	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	20 B	<DE8A847BFF8C343D69B85 3A215E6EE775EF2EF96>	Calculated hash. The length depends on the algorithm used.
5	1 C	<19>	Delimiter, indicating that there is a trailer for the message.
6	8 A	= 2EF96	Message trailer, as defined in the command.

JC Command - Translate a PIN Block from TPK to LMK Encryption.

This command takes a PIN block from a terminal encrypted using a TPK and returns an LMK-encrypted PIN. The LMK-encrypted PIN can be stored, or used as input to a number of commands which accept LMK-encrypted PINs (e.g. PIN mailers, or generate a PVV from an LMK-encrypted PIN).

The LMK-encryption of a PIN uses an algorithm involving the account number, such that the result will be different for two cardholders who have the same PIN.

In the example below, the TPK is single-length DES. The user has specified a header (with the length set to 4 characters in the host port settings) and a trailer.

Command

```
HDR5JCC8784A7C991251031166E5FDD7BCFCEA01787296260268<19>=41781
```

Field	Format	Value	Description
1	4 A	HDR5	Message header (as defined by user).
2	2 A	JC	Command Code.
3	16 H	C8784A7C99125103	The TPK (encrypted under LMK pair 14-15 variant 0 or LMK pair 36-37, depending on PCI HSM compliance settings) under which the PIN block is currently encrypted.
4	16 H	1166E5FDD7BCFCEA	The source PIN Block, encrypted under the TPK.
5	2 N	01	Source PIN Block format code 01 = ISO 9564-1 & ANSI X9.8 format 0
6	12 N	787296260268	Account Number. For PIN Block format code = 01, the 12 right-most digits of the PAN excluding the check digit.
7	1 C	<19>	End Message delimiter, to indicate that what follows is a message trailer.
8	6 A	=41781	Message trailer, as provided by the user.

Response

```
HDR5JD0041781<19>=41781
```

Field	Format	Value	Description
1	4 A	HDR5	Message header (as defined by user).
2	2 A	JD	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	5 N	41781	Output PIN encrypted under LMK pair 02-03. Notes: The length of the PIN is defined by the payShield 9000 security settings. The LMK encrypted PIN uses an algorithm involving the account number.
5	1 C	<19>	End Message delimiter, to indicate that what follows is a message trailer.
6	6 A	=41781	Message trailer, as provided in the command.

JE Command - Translate a PIN Block from ZPK to LMK Encryption.

The JE command performs a very similar function to the JC command described above, but in the case of JE the received PIN Block is coming from the payments network and is encrypted using a ZPK.

In the example below, the ZPK is a double-length TDES key. The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
HDR5JEU16FAD09093F342EC35AE6049B4F80834BE09A161870F9E8F0134567
8912345
```

Field	Format	Value	Description
1	4 A	HDR5	Message header (as defined by user).
2	2 A	JE	Command Code.
3	1 A + 32 H	U16FAD09093F342EC35AE6049B4F80834	The ZPK (encrypted under LMK pair 06-07) under which the PIN block is currently encrypted.
4	16 H	BE09A161870F9E8F	The source PIN Block, encrypted under the ZPK.
5	2 N	01	Source PIN Block format code 01 = ISO 9564-1 & ANSI X9.8 format 0
6	12 N	345678912345	Account Number. For PIN Block format code = 01, the 12 right-most digits of the PAN excluding the check digit.

Response

```
HDR5JF0081011
```

Field	Format	Value	Description
1	4A	HDR5	Message header (as defined by user).
2	2A	JF	Response Code.
3	2A	00	Error Code. 00 = No error.
4	5N	81011	Output PIN encrypted under LMK pair 02-03. Notes: The length of the PIN is defined by the payShield 9000 security settings. The LMK encrypted PIN uses an algorithm involving the account number.

JE Command – Translate a PIN from ZPK to LMK encryption (using a ZPK and PIN Block held in user storage)

In this example, both the LMK-encrypted PIN and the LMK-encrypted ZPK are held in user storage and are identified in the command using their user storage indexes. The ZPK is a double-length TDES key.

The LMK is of the variant TDES type.

Command

```
0260JEUK012K20001123456789012
```

Field	Format	Value	Description
1	4 A	0269	Message header (as defined by user).
2	2 A	JE	Command Code.
3	2 A + 3 H	UK012	Reference to the ZPK in user storage. The 'U' identifies the key scheme as double-length TDES using a variant LMK. 'K' is the index flag. '012' is the index for the ZPK in user storage.
4	'K' + 3 H	K200	Reference to the PIN Block held in user storage. 'K' is the index flag, and '200' is the index in user storage where the PIN Block is stored.
5	2 N	01	PIN Block format code
6	12 N	123456789012	The 12 right-most digits of the PAN, excluding the check digit.

Response

```
0260JF0082050
```

Field	Format	Value	Description
1	4 A	0260	Message header (as defined by user).
2	2 A	JF	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	5 N	82050	PIN encrypted under the LMK

JG Command - Translate a PIN from LMK to ZPK Encryption.

The JG command is the converse of the JE command described above. It takes an LMK-encrypted PIN and translates it into a PIN Block encrypted under a ZPK for transmission into the payments network.

In the example below, the ZPK is a single-length DES key. The user has specified a header (with the length set to 4 characters in the host port settings) and a trailer.

Command

```
HDR5 JG95FE090036A200C10178729626026841781<19>=FC96
```

Field	Format	Value	Description
1	4 A	HDR5	Message header (as defined by user).
2	2 A	JG	Command Code.
3	16 H	95FE090036A200C1	The ZPK that the PIN Block is to be encrypted under. The ZPK is single-length DES and is encrypted under LMK pair 06-07 variant 0.
4	2 N	01	The PIN Block format code for the ZPK-encrypted PIN. 01 = ISO 9564-1 & ANSI X9.8 format 0
5	12 N	78729626026841781	Account Number. For PIN Block format code = 01, the 12 right-most digits of the PAN excluding the check digit.
6	5 N	41781	The LMK-encrypted PIN, encrypted under LMK pair 02-03. Notes: The length of the PIN is defined by the payShield 9000 security settings. The LMK encrypted PIN uses an algorithm involving the account number.
7	1 C	<19>	End Message delimiter, to indicate that what follows is a message trailer.
8	5 A	=FC96	Message trailer, as provided by the user.

Response

```
HDR5JH00B6720B1B0DBFFC96<19>=FC96
```

Field	Format	Value	Description
1	4 A	HDR5	Message header (as defined by user).
2	2 A	JH	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	16 H	B6720B1B0DBFFC96	PIN Block, encrypted under the ZPK.
5	1 C	<19>	End Message delimiter, to indicate that what follows is a message trailer.
6	5 A	=FC96	Message trailer, as provided in the command.

JG Command – Translate a PIN from LMK to ZPK encryption (using a ZPK held in user storage)

In this example, the LMK-encrypted ZPK is held in user storage and is identified in the command using its user storage indexes. The ZPK is a double-length TDES key.

In this example, the security setting *User storage key length* has been set to **S**ingle. (Because the user storage length was set to Single, the key was loaded using the LA command into 2 contiguous blocks.)

The LMK is of the variant TDES type.

Command

```
0260 JGUK0120112345678901282050
```

Field	Format	Value	Description
1	4 A	0269	Message header (as defined by user).
2	2 A	JG	Command Code.
3	2 A + 3 H	UK012	Reference to the ZPK in user storage. The 'U' identifies the key scheme as double-length TDES using a variant LMK. 'K' is the index flag. '012' is the index for the ZPK in user storage.
4	2 N	01	PIN Block format code
5	12 N	123456789012	The 12 right-most digits of the PAN, excluding the check digit.
6	5 N	82050	PIN encrypted under the LMK

Response

```
0260JH0028D2BF543AC94665
```

Field	Format	Value	Description
1	4 A	0260	Message header (as defined by user).
2	2 A	JF	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	16 H	28D2BF543AC94665	PIN Block encrypted under the ZPK

LA Command - Load Data to User Storage (double-length TDES key encrypted under a variant LMK).

The LA command is used to load multiple blocks of data into the payShield 9000's internal user storage area. This process is described in this Application Note.

The data can be in blocks of 16, 32, or 48 hexadecimal digits (8, 16, or 24 Bytes) or (after v2.2a) of variable length – as specified in the security settings. Each block is identified by an index of 0-4,095 (0-FFF in hexadecimal).

In this example, the security setting *User storage key length* has been set to **S**ingle. We are storing two 8-byte blocks to indexes 0FE and 0FF representing a double-length TDES ZMK key. The user has specified a header (with the length set to 4 characters in the host port settings) and a trailer.

Command

```
HDRALAK0000292385025801691228682054947200919<19>=01
```

Field	Format	Value	Description
1	4 A	HDRA	Message header (as defined by user).
2	2 A	LA	Command Code.
3	1 A	K	Index Flag
4	3 H	0FE	Index for location where first data block in the command is to be stored.
5	2 H	02	Number of data blocks included in the command
6	16 H	9238502580169122	Data Block 1. (The data block size has been set to "Single" in the *CS* console command or HSM Manager *Edit/General Settings* dialogue box.)
7	16 H	8682054947200919	Data Block 2. (The data block size has been set to "Single" in the *CS* console command or HSM Manager *Edit/General Settings* dialogue box.)
8	1 C	<19>	End Message delimiter, to indicate that what follows is a message trailer.
9	4 A	=01	Message trailer, as provided by the user.

Response

```
HDRALB00<19>=01
```

Field	Format	Value	Description
1	4 A	HDRA	Message header (as defined by user).
2	2 A	LB	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	1 C	<19>	End Message delimiter, to indicate that what follows is a message trailer.
5	4 A	=01	Message trailer, as provided in the command.

LA Command - Load Data to User Storage (RSA private key encrypted under a TDES variant LMK).

In this example, the security setting *User storage key length* has been set to **V**ariable: this setting is not available in payShield 9000 v2.2a or earlier. We are storing a variant LMK-encrypted RSA private key consisting of 176 bytes of binary data.

Command

```
0000LABK0000176<813C82DB56E9AEA2A4AFC0B068259553DF18C32AFCA60F
49833F52CFC0E6FC70287794AEE01F43F96B91B0492229367B6C9DDC2715CD
668F94880D0AC0C4E5A0B22C8DAF12E2DA2F9BF8F475959D64FFCF80F03F86
DAF297B92A2F1E0CE1CF629564EADB3B5689A0FAE636B9B6E27E652186A45B
C49CB13307281F5B3A63FBA315047071CDDBE15247E278120C467481CF99F4
EE09626E85613B1994B8230FED7F4C04B8BC9205A6CFB285B1DBA22D24>
```

Field	Format	Value	Description
1	4 A	0000	Message header (as defined by user).
2	2 A	LA	Command Code.
3	1A	B	Index Flag (B = Binary).
4	3 H	000	Index for location where the data block is to be stored. Because this in the range 000-07F, up to 1,000 bytes can be stored.
5	4N	0176	Total length of data, in bytes.
6	176B	<813C82DB56E9AEA2A4AFC 0B068259553DF18C32AFCA … 994B8230FED7F4C04B8BC9 205A6CFB285B1DBA22D24>	RSA Private key encrypted under the variant LMK.

Response

```
0000LB00
```

Field	Format	Value	Description
1	4 A	0000	Message header (as defined by user).
2	2 A	LB	Response Code.
3	2 A	00	Error Code. 00 = No error.

LA Command - Load Data to User Storage (RSA private key encrypted under a TDES key block LMK).

In this example, the security setting *User storage key length* has been set to **V**ariable: this setting is not available in payShield 9000 v2.2a or earlier. We are storing an RSA private key consisting of 336 bytes of binary data held with a 376-byte key block.

Command

```
0000LABK02203760037603RN00N02020005TPB0B4vzoOWb<F343F5C367CB55
7BC217195EBFD4821E91E70D05CADF74C2E9F90F4E4184B9D8827F03CB337A
0873787F00D27FED0D1CF71AF4391DE2E4101E47D602BB5D5C3E91E54BBE03
4F19A4AF3562766AC607940FE58F66EEFDE42D1D904C1510B1BB6323DA3505
17EFA647215A4224874F24C6A1A4A7E4FEA6DB93640AABA4B7A3F2E7923C21
526A4C2B69FF254F4C6DF633103475404CF90A8FC2C6ADF4187FBFC1414D79
575275353751B4E1777CE093347A128AB30A959BE8827433E522147C8129E6
6BC59BD4466E08B9BCBE136E6FDB00EB6F12B81957AA37544800D85CAFE630
894302E92F84ADD2C9A22F302A6CD772A47EBF9BEBF82C4718321100D73A66
991277309CD3D37D6377180336BF9F1FE81303A6C33EBD1B29A9A4530DDD54
D8592D288397BBA7F06AFA80AE6E28FA0788463B328D9E6FEC8CFFCD8BAE2C
E2E904F09E3AD2CFCE4B045CEC4D2C23061455>99653B1A
```

Field	Format	Value	Description
1	4 A	0000	Message header (as defined by user).
2	2 A	LA	Command Code.
3	1A	B	Index Flag (B = Binary).
5	3 H	022	Index for location where the data block is to be stored. Because this in the range 000-07F, up to 1,000 bytes can be stored.
5	4N	0376	Total length of data, in bytes.
6.1	1N	0	(Key Block Header) Version ID
6.2	4N	0376	(Key Block Header) Total length of key block
6.3	2H	03	(Key Block Header) Key Usage (03 = RSA Private Key)
6.4	1A	R	(Key Block Header) Algorithm (R = RSA)
6.5	1A	N	(Key Block Header) Mode of Use (N = No special restrictions)
6.6	2N	00	(Key Block Header) Key Version No.
6.7	1A	N	(Key Block Header) Exportability (N = No exportability)
6.8	2N	02	(Key Block Header) Number of optional header blocks
6.9	2N	02	(Key Block Header) LMK ID.
6.10	2A	00	(Optional Key Block Header 1) Block Type (00 = Key Status)
6.11	2H	05	(Optional Key Block Header 1) Block Length
6.12	1A	L	(Optional Key Block Header 1) Key Status (L = Live)
6.13	2A	PB	(Optional Key Block Header 2) Block Type (PB = Padding Block)
6.14	2H	0B	(Optional Key Block Header 2) Block Length
6.15	7A	4vzoOWb	(Optional Key Block Header 2) Padding text
6.16	376B	<F343F5C367CB557BC2171 95EBFD4821E91E70D05CAD … 904F09E3AD2CFCE4B045CE C4D2C23061455>	(Key Block data) RSA Private key encrypted under the key block LMK
6.17	8A	99653B1A	Key Block MAC

In this example, Field 6, the data field in the LA command, has been subdivided to explain the key block structure.

Response

```
0000LB00
```

Field	Format	Value	Description
1	4 A	`0000`	Message header (as defined by user).
2	2 A	`LB`	Response Code.
3	2 A	`00`	Error Code. `00` = No error.

LE Command - Read Data from User Storage (single/double/triple block size setting).

With the LE command, data previously written to the payShield 9000's internal user storage using the LA command can be retrieved.

In this example, the security setting *User storage key length* has been set to **S**ingle. We are retrieving two 8-byte blocks from indexes 0FE and 0FF representing a double-length TDES ZMK key. The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
HDRBLEK0FE02<19>=09199238502580169122
```

Field	Format	Value	Description
1	4 A	HDRB	Message header (as defined by user).
2	2 A	LE	Command Code.
3	1 A	K	Index Flag
4	3 H	0FE	Index for location where first data block is stored.
5	2 H	02	Number of data blocks to be retrieved
6	1 C	<19>	End Message delimiter, to indicate that what follows is a message trailer.
7	21 A	=09199238502580169122	Message trailer, as provided by the user.

Response

```
HDRBLF0086820549472009199238502580169122<19>=0919
9238502580169122
```

Field	Format	Value	Description
1	4 A	HDRB	Message header (as defined by user).
2	2 A	LF	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	16 H	8682054947200919	Data Block 1. (The data block size has been set to “Single” in the *CS* console command or HSM Manager *Edit/General Settings* dialogue box.)
5	16 H	9238502580169122	Data Block 2. (The data block size has been set to “Single” in the *CS* console command or HSM Manager *Edit/General Settings* dialogue box.)
6	1 C	<19>	End Message delimiter, to indicate that what follows is a message trailer.
7	21 A	=09199238502580169122	Message trailer, as provided in the command.

LE Command - Read Data from User Storage (variable block size setting).

In this example, the security setting *User storage key length* has been set to **V**ariable – this capability is not available in payShield 9000 version 2.2a or earlier. When this option is selected, a single data block is retrieved.

We are retrieving an RSA private key encrypted using a key block LMK. The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
0000LEK022
```

Field	Format	Value	Description
1	4 A	0000	Message header (as defined by user).
2	2 A	LE	Command Code.
3	1 A	K	Index Flag (Note: this field always has a value of 'K', irrespective of the value of the Index Flag used when the data was read into user storage using the LA command.)
4	3 H	022	Index for location where first data block is stored.

Response

```
0000LF00B03760037603RN00N02020005TPB0B4vzoOWb<F343F5C367CB557B
C217195EBFD4821E91E70D05CADF74C2E9F90F4E4184B9D8827F03CB337A08
73787F00D27FED0D1CF71AF4391DE2E4101E47D602BB5D5C3E91E54BBE034F
19A4AF3562766AC607940FE58F66EEFDE42D1D904C1510B1BB6323DA350517
EFA647215A4224874F24C6A1A4A7E4FEA6DB93640AABA4B7A3F2E7923C2152
6A4C2B69FF254F4C6DF633103475404CF90A8FC2C6ADF4187FBFC1414D7957
5275353751B4E1777CE093347A128AB30A959BE8827433E522147C8129E66B
C59BD4466E08B9BCBE136E6FDB00EB6F12B81957AA37544800D85CAFE63089
4302E92F84ADD2C9A22F302A6CD772A47EBF9BEBF82C4718321100D73A6699
1277309CD3D37D6377180336BF9F1FE81303A6C33EBD1B29A9A4530DDD54D8
592D288397BBA7F06AFA80AE6E28FA0788463B328D9E6FEC8CFFCD8BAE2CE2
E904F09E3AD2CFCE4B045CEC4D2C23061455>99653B1A
```

Field	Format	Value	Description
1	4 A	0000	Message header (as defined by user).
2	2 A	LF	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	1A	B	Data type (B = Binary).
5	4N	0376	Total length of data, in bytes.
6	376B	0037603RN00N02020005TP B0B4vzoOWb<F343F5C367C B557BC217195EBFD4821E9 … FCE4B045CEC4D2C2306145 5>99653B1A	Key Block, including RSA Private key encrypted under LMK

M0 Command – encrypt a block of data (using a key block LMK-included in the command).

The M0 command is used for encryption of general messages. The encryption key may be provided as one of the command's parameters, as in this example. It is a triple-length TDES key encrypted using a key block LMK. The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
0000M00000FFFS1009621AB00N00021E96DD97D8B0F388B525E93F522952F7
A394F29D6E8C20D8F2775DF5654BDCAFCA96DD6F427AA8630040<8D56A1282
943E097423A6FA8A951BD127D94F16088FEE8DD1BFC506B22CCE4F022BDB3F
63A2E6C531F29EFB59130274827CDCE094BD84295B68EE4227B610738>
```

Field	Format	Value	Description
1	4 A	0000	Message header (as defined by user).
2	2 A	M0	Command Code.
3	2 N	00	Mode flag. (00 = ECB)
4	1 N	0	Input format (0 = Binary)
5	1 N	0	Output format (0 = Binary)
6	3 H	FFF	Key type. (Always set to FFF when using a key block LMK.)
7	1 A + 48 A	S1009621AB00N00021E96DD97D8B0F388B525E93F522952F7A394F29D6E8C20D8F2775DF5654BDCAFCA96DD6F427AA863	Key to be used to encrypt the message. The 'S' at the start of the field is the key scheme and identifies that the key is encrypted using a key block LMK. The 16-ASCII character header information defines the following: 1 Version 1 (AES) 0096 Total length 21 Key usage (DEK) A Algorithm (AES) B Mode of use (encrypt & decrypt) 00 Key version no. N No export allowed 00 No. of optional blocks 02 LMK ID
8	4 H	0040	Length of message to be encrypted (i.e. 64 in decimal)
9	64 B	<8D56A1282943E097423A6FA8A951BD127D94F16088F … 59130274827CDCE094BD84295B68EE4227B610738>	Message to be encrypted.

Response

```
0000M1000040<82DA2F20A0121F58EA006E47AB6321BD09A592AC3C1103206
9E34C1011261F4128BD73CF3BDBE5C91515FF65928FB812380F58B722DA114
9A464E77679CD60F0>
```

Field	Format	Value	Description
1	4 A	0000	Message header (as defined by user).
2	2 A	M1	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	4 H	0040	Length of encrypted message (i.e. 64 in decimal)
5		<82DA2F20A0121F58EA006 E47AB6321BD09A592AC3C1 … 5928FB812380F58B722DA1 149A464E77679CD60F0>	Encrypted message

M0 Command – encrypt a block of data (using a key block LMK-encrypted key held in user storage).

The M0 command is used for encryption of general messages. The encryption key may be provided as one of the command's parameters or, as in this example, the command may contain a reference to the key held in the user storage area.

In this example, the security setting *User storage key length* has been set to **V**ariable – this capability is not available in payShield 9000 version 2.2a or earlier. When this option is selected, a single data block is retrieved.

We are retrieving from user storage a TDES encryption key encrypted using a key block LMK. The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
0000M00000FFFSK0110040<8D56A1282943E097423A6FA8A951BD127D94F16
088FEE8DD1BFC506B22CCE4F022BDB3F63A2E6C531F29EFB59130274827CDC
E094BD84295B68EE4227B610738>
```

Field	Format	Value	Description
1	4 A	0000	Message header (as defined by user).
2	2 A	M0	Command Code.
3	2 N	00	Mode flag. (00 = ECB)
4	1 N	0	Input format (0 = Binary)
5	1 N	0	Output format (0 = Binary)
6	3 H	FFF	Key type. (Always set to FFF when using a key block LMK.)
7	2 A + 3 H	SK011	Index to key in user storage. ('S' identifies key scheme as key block LMK; 'K' is the index flag; '011' is the index to the key in user storage. (Note: the index flag always has a value of 'K', irrespective of the value of the Index Flag used when the data was read into user storage using the LA command.))
8	4 H	0040	Length of message to be encrypted (i.e. 64 in decimal)
9	64 B	<8D56A1282943E097423A6 FA8A951BD127D94F16088F … 59130274827CDCE094BD84 295B68EE4227B610738>	Message to be encrypted.

Response

```
0000M1000040<82DA2F20A0121F58EA006E47AB6321BD09A592AC3C1103206
9E34C1011261F4128BD73CF3BDBE5C91515FF65928FB812380F58B722DA114
9A464E77679CD60F0>
```

Field	Format	Value	Description
1	4 A	0000	Message header (as defined by user).
2	2 A	M1	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	4 H	0040	Length of encrypted message (i.e. 64 in decimal)
5		<82DA2F20A0121F58EA006 E47AB6321BD09A592AC3C1 … 5928FB812380F58B722DA1 149A464E77679CD60F0>	Encrypted message

M2 Command - Decrypt Data

The M2 command is used to decrypt data which has been encrypted using a DES or TDES key.

In the example below, the data has been encrypted by a DUKPT terminal. The BDK (identified in the KSN returned by the terminal) is a double-length TDES BDK. The data encryption key is derived by the payShield 9000 using the BDK supplied by the host system and the KSN provided by the terminal.

The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
HDR3M20102609U1B42F9A78E06701748394348 90D0D58F609FFFF02016E365
8200003000000000000000000020<95E55D4EC345B87E7878C9CCFA0099287D
B8908D21FEA8E0CB9A2BAF20C2FB94>
```

Field	Format	Value	Description
1	4 A	HDR3	Message header (as defined by user).
2	2 A	M2	Command Code.
3	2 N	01	Mode Flag. 01 = CBC
4	1 N	0	Input Format Flag. 0 = Binary
5	1 N	2	Output Format Flag. 2 = Text
6	3 H	609	Key Type. 609 = Type 2 BDK, encrypted under Variant LMK pair 28-29, variant 6.
7	1 A + 32 H	U1B42F9A78E0670174839434890D0D58F	Key that data is encrypted under. U = Double-length TDES key scheme The remainder of the field is the Type 2 BDK encryption key encrypted under the LMK.
8	3 H	609	KSN Descriptor. 6 = BDK identifier length (in Hex chars.) 0 = Sub-key identifier length (in Hex chars.) 9 = Device identifier length (in Hex chars.)
9	20 H	FFFF02016E3658200003	KSN (Key Serial Number) FFFF02 = BDK identifier 016E36582 = Device identifier 00003 = Transaction Counter
10	16 H	0000000000000000	IV (Initial Value)
11	4 H	0020	Message Length, i.e. length (in hexadecimal) of message to be decrypted, in bytes.
12	32 Bytes	<95E55D4EC345B87E7878C9CCFA0099287DB8908D21FEA8E0CB9A2BAF20C2FB94>	Message to be decrypted.

Response

```
HDR3M300CB9A2BAF20C2FB9400200123456789123456<F04C6352435B49FDD
4E161769857233D>
```

Field	Format	Value	Description
1	4 A	HDR3	Message header (as defined by user).
2	2 A	M3	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	16 H	CB9A2BAF20C2FB94	Output Initial Value (IV)
5	4 H	0020	Message Length, i.e. length (in hexadecimal) of decrypted message, in bytes.
6	32 Bytes	0123456789123456<F04C6352435B49FDD4E161769857233D>	Decrypted message.

M4 Command – Translate a Data Block

The M4 command is used to take a data block encrypted under a source key and output the data block encrypted under a destination key.

In this example, the user has specified that no headers or trailers will be used.

Command

```
M400000000AU58C121B3ED17D080DA32AC8ECECFE2B900AUB2108640409BB3
D72FA549E40539BB110018<33385B9D0DB4521EA31D3644C0BEFBFE424686D
24C96CBB6>
```

Field	Format	Value	Description
1	2 A	M4	Command Code.
2	2 N	00	Source Mode Flag. 00 = ECB (does not require an IV)
3	2 N	00	Destination Mode Flag. 00 = ECB (does not require an IV)
4	1 N	0	Input Format Flag. 0 = Binary
5	1 N	0	Output Format Flag. 0 = Binary
6	3 H	00A	Source Key Type 00A = ZEK
7	'U' + 32 H	U58C121B3ED17D080DA32AC8ECECFE2B9	Source key (double-length DES)
8	3 H	00A	Destination Key Type 00A = ZEK
9	'U' + 32 H	UB2108640409BB3D72FA549E40539BB11	Destination key (double-length DES)
10	4 H	0018	Message length (i.e. 24 in decimal)
11	24 B	<33385B9D0DB4521EA31D3644C0BEFBFE424686D24C96CBB6>	The source key-encrypted message that is to be encrypted under the destination key.

Response

```
M5000018<418B11480E5791484633D44C3AE7AB83EC72F588ADF70F05>
```

Field	Format	Value	Description
1	2 A	M5	Response Code.
2	2 A	00	Error Code. 00 = No error.
3	4 N	0018	Message length (i.e. 24 in decimal)
4	24 B	<418B11480E5791484633D44C3AE7AB83EC72F588ADF70F05>	The message encrypted under the destination key

M6 Command – Generate a MAC

The M6 command calculates a MAC on a block of data, which may be one of a sequence of data blocks.

In this example, the user has specified that a header but no trailer will be used.

Command

```
0000M601012008U050743F67FBFD7D56562194469B344D9003074212B3C32B
BFAE5B21D21ABF3A47209FDA657F0D3A90C3A
```

Field	Format	Value	Description
1	4 A	0000	Header (as provided by the user).
2	2 A	M6	Command Code.
3	1 N	0	Mode Flag. 0 = This is the one and only block of data that the MAC is to be calculated over.
4	1 N	1	Input Format Flag. 1 = Hex-encoded Binary
5	1 N	0	MAC size. 0 = 8 hex digits.
6	1 N	1	MAC algorithm. 1 = ISO 9797 MAC algorithm 1 (= ANSI X9.9 when used with a single-length key) (DES only)
7	1 N	2	Padding method. 2 = ISO 9797 Padding Method 2.
8	3 H	008	Key type. 008 = ZAK
9	1 A + 32 H	U050743F67FBFD7D56562194469B344D9	Key to be used in calculating the MAC. (This is a double-length DES key.)
10	4 H	0030	Message length (i.e. 38 in decimal).
11	48 H	74212B3C32BBFAE5B21D21ABF3A47209FDA657F0D3A90C3A	Message that the MAC is to be calculated over.

Response

```
0000M70022C7A93A
```

Field	Format	Value	Description
1	4 A	0000	Message header (as defined by user).
2	2 A	M7	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	8 H	4B72B3CB	The calculated MAC.

M8 Command – Verify a MAC

The M8 command verifies a MAC that has previously been calculated on a block of data, which may be one of a sequence of data blocks – for example by using the M6 command.

In this example, the user has specified that a header but no trailer will be used.

Command

```
0000M800010008UBEFAD634C0F5B85EDCF338F8B640F48F0018<E8C3C44FF5
BA508FBA491E938CC1E2C9AE605D47BDD50908>2E311B62
```

Field	Format	Value	Description
1	4 A	0000	Header (as provided by the user).
2	2 A	M8	Command Code.
3	1 N	0	Mode Flag. 0 = This is the one and only block of data that the MAC is to be calculated over.
4	1 N	0	Input Format Flag. 0 = Binary
5	1 N	0	MAC size. 0 = 8 hex digits.
6	1 N	1	MAC algorithm. 1 = ISO 9797 MAC algorithm 1 (= ANSI X9.9 when used with a single-length key) (DES only)
7	1 N	0	Padding method. 0 = No padding.
8	3 H	008	Key type. 008 = ZAK
9	1 A + 32 H	UBEFAD634C0F5B85EDCF338F8B640F48F	Key to be used in calculating the MAC. (This is a double-length TDES key.)
10	4 H	0018	Message length (i.e. 24 in decimal).
11	48 H	<E8C3C44FF5BA508FBA491E938CC1E2C9AE605D47BDD50908>	Message that the MAC is to be calculated over.
12	8 H	2E311B62	The MAC to be verified

Response

```
0000M900
```

Field	Format	Value	Description
1	4 A	0000	Message header (as defined by user).
2	2 A	M9	Response Code.
3	2 A	00	Error Code. 00 = No error. Other codes would give the reason for a failure of the MAC verification.

MY Command – Verify and Translate a MAC (first data block)

The MY command is used to verify a MAC previously generated (e.g. using the M6 host command). If the verification is successful, the command generates a new MAC on the same message but using a different TAK (Terminal Authentication Key) or ZAK (Zone Authentication Key).

The data block is the first of a multi-block series.

The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
0000MY11030003UAA991FAD71CDB654668FC65ADF6F5F2B030003UA2373C83
5D758A505AAFA9356DF4B92200409B8C2DBD52685F3B77B40096A210F2CA91
7B9FD15174EBF6BF2041A5DF7C24B9
```

Field	Format	Value	Description
1	4 A	0000	Message header (as defined by user).
2	2 A	MY	Command Code.
3	1 N	1	Mode Flag. 1 = First block of a multi-block message.
4	1 N	1	Input Format Flag. 1 = Hex-encoded Binary
5	1 N	0	Source MAC size. 0 = 8 Hex digits
6	1 N	3	Source MAC algorithm. 1 = ISO 9797 MAC algorithm 3 (= ANSI X9.9 when used with a double-length key) (DES only)
7	1 N	0	Source Padding Method. 0 = No padding.
8	3 H	003	Source Key Type. 003 = TAK (Terminal Authentication Key).
9	1 A + 32 H	UAA991FAD71CDB654668FC65ADF6F5F2B	Source Key. (This is a double-length TDES key.)
10	1 N	0	Destination MAC size. 0 = 8 Hex digits
11	1 N	3	Destination MAC algorithm. 1 = ISO 9797 MAC algorithm 3 (= ANSI X9.9 when used with a double-length key) (DES only)
12	1 N	0	Destination Padding Method. 0 = No padding.
13	3 H	003	Destination Key Type. 003 = TAK (Terminal Authentication Key).
14	1 A + 32 H	UA2373C835D758A505AAFA9356DF4B922	Destination Key. (This is a double-length TDES key.)
15	4 H	0040	The length of the following message field (i.e. 64 decimal).
16	64 H	9B8C2DBD52685F3B77B40096A210F2CA917B9FD15174EBF6BF2041A5DF7C24B9	The message over which the MAC is to be verified and re-calculated.

Response

```
0000MZ007AC27911D5DE42207EB6C5E5E79DDCB3
```

Field	Format	Value	Description
1	4 A	`0000`	Message header (as defined by user).
2	2 A	`MZ`	Response Code.
3	2 A	`00`	Error Code. 00 = No error. Other codes would give reason for verification failure.
4	16 H	`7AC27911D5DE4220`	Source Initial Value (IV) calculated using Source Key, for input with the next data block.
5	16 H	`7EB6C5E5E79DDCB3`	Destination Initial Value (IV) – calculated using Destination key, for input with the next data block.

MY Command – Verify and Translate a MAC (middle data block)

The MY command is used to verify a MAC previously generated (e.g. using the M6 host command). If the verification is successful, the command generates a new MAC on the same message but using a different TAK (Terminal Authentication Key) or ZAK (Zone Authentication Key).

The data block is the second of a multi-block series.

The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
0001MY21030003UAA991FAD71CDB654668FC65ADF6F5F2B030003UA2373C83
5D758A505AAFA9356DF4B9227AC27911D5DE42207EB6C5E5E79DDCB30040EC
43B0FF5886E4C6F68B32111986DB46CB32D1300CD52FAE79AD09388F1EAE1F
```

Field	Format	Value	Description
1	4 A	0001	Message header (as defined by user).
2	2 A	MY	Command Code.
3	1 N	2	Mode Flag. 2 = Middle block of a multi-block message.
4	1 N	1	Input Format Flag. 1 = Hex-encoded Binary
5	1 N	0	Source MAC size. 0 = 8 Hex digits
6	1 N	3	Source MAC algorithm. 1 = ISO 9797 MAC algorithm 3 (= ANSI X9.9 when used with a double-length key) (TDES only)
7	1 N	0	Source Padding Method. 0 = No padding.
8	3 H	003	Source Key Type. 003 = TAK (Terminal Authentication Key).
9	1 A + 32 H	UAA991FAD71CDB654668FC65ADF6F5F2B	Source Key. (This is a double-length TDES key.)
10	1 N	0	Destination MAC size. 0 = 8 Hex digits
11	1 N	3	Destination MAC algorithm. 1 = ISO 9797 MAC algorithm 3 (= ANSI X9.9 when used with a double-length key) (TDES only)
12	1 N	0	Destination Padding Method. 0 = No padding.
13	3 H	003	Destination Key Type. 003 = TAK (Terminal Authentication Key).
14	1 A + 32 H	UA2373C835D758A505AAFA9356DF4B922	Destination Key. (This is a double-length TDES key.)
15	16 H	7AC27911D5DE4220	Source IV (provided in output from preceding data block).
16	16 H	7EB6C5E5E79DDCB3	Destination IV (provided in output from preceding data block).
17	4 H	0040	The length of the following message field (i.e. 64 decimal).

Field	Format	Value	Description
18	64 H	EC43B0FF5886E4C6F68B32 111986DB46CB32D1300CD5 2FAE79AD09388F1EAE1F	The message over which the MAC is to be verified and re-calculated.

Response

```
0001MZ00E89F8FE95CB12A44043130D99F9578A4
```

Field	Format	Value	Description
1	4 A	0001	Message header (as defined by user).
2	2 A	MZ	Response Code.
3	2 A	00	Error Code. 00 = No error. Other codes would give reason for verification failure.
4	16 H	E89F8FE95CB12A44	Source Initial Value (IV) calculated using Source Key, for input with the next data block.
5	16 H	043130D99F9578A4	Destination Initial Value (IV) – calculated using Destination key, for input with the next data block.

MY Command – Verify and Translate a MAC (last data block)

The MY command is used to verify a MAC previously generated (e.g. using the M6 host command). If the verification is successful, the command generates a new MAC on the same message but using a different TAK (Terminal Authentication Key) or ZAK (Zone Authentication Key).

The data block is the third and last of a multi-block series.

The user has specified a header (with the length set to 4 characters in the host port settings), but no trailer.

Command

```
0002MY31030003UAA991FAD71CDB654668FC65ADF6F5F2B030003UA2373C83
5D758A505AAFA9356DF4B922E89F8FE95CB12A44043130D99F9578A40040A2
0A7E4D8E8CE1F82881D222E1360B14EDDECB2ED02F1A20CABA45BCF64EF5A6
3D6253F6
```

Field	Format	Value	Description
1	4 A	0002	Message header (as defined by user).
2	2 A	MY	Command Code.
3	1 N	3	Mode Flag. 3 = Last block of a multi-block message.
4	1 N	1	Input Format Flag. 1 = Hex-encoded Binary
5	1 N	0	Source MAC size. 0 = 8 Hex digits
6	1 N	3	Source MAC algorithm. 1 = ISO 9797 MAC algorithm 3 (= ANSI X9.9 when used with a double-length key) (TDES only)
7	1 N	0	Source Padding Method. 0 = No padding.
8	3 H	003	Source Key Type. 003 = TAK (Terminal Authentication Key).
9	1 A + 32 H	UAA991FAD71CDB654668FC65ADF6F5F2B	Source Key. (This is a double-length TDES key.)
10	1 N	0	Destination MAC size. 0 = 8 Hex digits
11	1 N	3	Destination MAC algorithm. 1 = ISO 9797 MAC algorithm 3 (= ANSI X9.9 when used with a double-length key) (TDES only)
12	1 N	0	Destination Padding Method. 0 = No padding.
13	3 H	003	Destination Key Type. 003 = TAK (Terminal Authentication Key).
14	1 A + 32 H	UA2373C835D758A505AAFA9356DF4B922	Destination Key. (This is a double-length TDES key.)
15	16 H	E89F8FE95CB12A44	Source IV (provided in output from preceding data block).
16	16 H	043130D99F9578A4	Destination IV (provided in output from preceding data block).
17	4 H	0040	The length of the following message field (i.e. 64 decimal).

Field	Format	Value	Description
18	64 H	A20A7E4D8E8CE1F82881D222E1360B14EDDECB2ED02F1A20CABA45BCF64EF5A6	The message over which the MAC is to be verified and re-calculated.
19	8 H	3D6253F6	The source MAC to be verified.

Response

```
0002MZ0025ED31B2
```

Field	Format	Value	Description
1	4 A	0002	Message header (as defined by user).
2	2 A	MZ	Response Code.
3	2 A	00	Error Code. 00 = No error. Other codes would give reason for verification failure.
4	8 H	25ED31B2	Destination MAC, generated using the destination key.

NG Command - Decrypt PIN from LMK-encryption to clear

The command is used when printing PIN mailers. It converts an LMK-encrypted PIN to cleartext. LMK encryption of the PIN uses an algorithm which involves the account number.

When the cleartext PIN is returned, the account number is returned in the form of a random reference number, such that PIN and account number do not appear together in the clear.

The command is available only:

- If enabled by the security setting *Enable clear PINs*.
- If the command is authorized.

Command

```
HDR6NG34567891234581011
```

Field	Format	Value	Description
1	4 A	HDR6	Message header (as defined by user).
2	2 A	NG	Command Code.
3	12 N	345678912345	Account Number.
4	5 N	81011	PIN encrypted under LMK pair 02-03. Notes: • The length of the encrypted PIN is defined by the payShield 9000 security settings. • The LMK encrypted PIN uses an algorithm involving the account number.

Response

```
HDR6NH001234F511599955529
```

Field	Format	Value	Description
1	4 A	HDR6	Message header (as defined by user).
2	2 A	NH	Response Code.
3	2 A	00	Error Code. 00 = No error.
4	5 H	1234F	The clear PIN, left-justified and padded with F.
5	12 N	511599955529	The reference number derived by encrypting the account number under the LMK. This allows the PIN to be associated with an account number without both being available in the clear.

PM Command - Verify Dynamic CVV/CVC (MasterCard)

The PM command allows a Dynamic CVV/CVC to be verified when processing a transaction. In this example, we are verifying a CVC3 in the magstripe track data of a MasterCard PayPass card.

Command

```
HDREPM12U42F5FBDFE936A8CCFC0F0CDBA7D80C03A5241060000000069;000
19<5241060000000069D13052020000000000003F>0000000083900032XX942
```

Field	Format	Value	Description
1	4 A	HDRE	Message header (as defined by user).
2	2 A	PM	Command Code.
3	1 N	1	Scheme ID. 1 = MasterCard PayPass
4	1 N	2	Version 2 = For MasterCard PayPass, PSN is provided in input and IVCVC3 calculated from provided mag. Stripe data.
5	1 A + 32 H	U42F5FBDFE936A8CCFC0F0CDBA7D80C03	Master Key for Dynamic CVV (MK-DCVV), encrypted under LMK pair 28-29 variant 7, where Scheme ID = 1. U = Double-length TDES key scheme
6	1 A	A	Key derivation method. A = EMV4.1 Book 2 Option A method.
7	16 N	5241060000000069	PAN. (Maximum length allowed for Scheme ID = 1 is 19 digits.)
8	1 A	;	Delimiter to indicate end of PAN.
9	2 N	00	PAN Sequence Number (PSN). 00 = not available.
10	3 N	019	Length of Track Data field. (REMINDER: we are using Scheme ID = 1, Version = 2 in this example.)
11	19 Bytes	<5241060000000069D13052020000000000003F>	Static Track (1 or 2) Data
12	10 N	0000000839	Unpredictable number. Random number provided to the card by the terminal during a PayPass transaction. (NOTE: the Unpredictable number could also use a format of 8D, e.g. 00000347.)
13	5 N	00032	Decimal value of Application Transaction Counter.
14	5 A	XX942	The CVC3 to be validated. The "X" characters are like wildcard characters such that, for example, a CVC3 of "XX942" will match a calculated CVC3 of "87942".

Response

```
HDREPN00
```

Field	Format	Value	Description
1	4 A	HDRE	Message header (as defined by user).
2	2 A	PN	Response Code.
3	2 A	00	Error Code. 00 = No error - i.e. verification successful.

PM Command - Verify Dynamic CVV/CVC (Visa)

In this example, we are verifying a dCVV in the magstripe track data of a Visa contactless card.

Command

```
HDREPM00UDD4BB1AF0B96841182E94DD953FF4146A13123456784808;06061
23005530598
```

Field	Format	Value	Description
1	4 A	HDRE	Message header (as defined by user).
2	2 A	PM	Command Code.
3	1 N	0	Scheme ID. 0 = Visa
4	1 N	0	Version Always 0 for Visa.
5	1 A + 32 H	UDD4BB1AF0B96841182E94DD953FF4146	Master Key for Dynamic CVV (MK-DCVV), encrypted under LMK pair 28-29 variant 7, where Scheme ID = 1. U = Double-length TDES key scheme
6	1 A	A	Key derivation method. A = EMV4.1 Book 2 Option A method.
7	14 N	13123456784808	PAN. (Maximum length allowed for Scheme ID = 0 is 16 digits.)
8	1 A	;	Delimiter to indicate end of PAN.
9	4N	0606	Card Expiry Date
10	3N	123	Service Code, from card's Track 2 data.
11	6N	005530	Application Transaction Counter (ATC) – padded to the left with 0's.
12	3N	598	dCVV to be validated

Response

```
HDREPN00
```

Field	Format	Value	Description
1	4 A	HDRE	Message header (as defined by user).
2	2 A	PN	Response Code.
3	2 A	00	Error Code. 00 = No error - i.e. verification successful.

THALES

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 <