

CryptoApplet User's Guide

Introduction

CryptoApplet is a simple low-level applet that allows you to use Cipher, Signature, and Message Digest algorithms as well as the on-board key generation available in the card. This applet was specially developed for the GemXpresso 211 PK card features, and of course the GSE in GXP211_PK or GXP211_PK_IS card simulation mode.

Command Set

Class known value

CLA_TEST	0x90
----------	------

Instruction known values

Cipher-defined instructions

INS_DES_ECB_ENC	0x20
INS_DES_ECB_DEC	0x22
INS_DES_CBC_ENC	0x24
INS_DES_CBC_DEC	0x26

INS_RSA_ENCRYPT	0xA0
INS_RSA_DECRYPT	0xA2
INS_RSA_DECRYPT_CRT	0xA4

Signature-defined instructions

INS_DES_MAC4_SIGN	0x30
INS_DES_MAC4_VERIFY_INIT	0x32
INS_DES_MAC4_VERIFY	0x34
INS_DES_MAC8_SIGN	0x36
INS_DES_MAC8_VERIFY_INIT	0x38
INS_DES_MAC8_VERIFY	0x3A

INS_RSA_SHA1_SIGN	0xB0
INS_RSA_MD5_SIGN	0xB2
INS_RSA_SHA1_VERIFY_INIT	0xB4
INS_RSA_SHA1_VERIFY	0xB6
INS_RSA_MD5_VERIFY	0xB8

Message Digest-defined instructions

INS_SHA1	0xD0
INS_MD5	0xD2

Key initialization-defined instructions

INS_PUT_DES_KEY	0xE0
INS_PUT_DES3_KEY	0xE2

INS_PUT_PRIVATE_RSA_KEY	0xF0
INS_PUT_CRT_PRIVATE_RSA_KEY	0xF2
INS_PUT_PUBLIC_RSA_KEY	0xF4

Key generation-defined instructions

INS_GENERATE_RSA_KEY	0xF6
INS_GENERATE_RSA_CRT_KEY	0xF8

Key retrieval-defined instructions

INS_GET_PRIVATE_RSA_KEY	0xFA
INS_GET_CRT_PRIVATE_RSA_KEY	0xFC
INS_GET_PUBLIC_RSA_KEY	0xFE

Install Parameter

Details

The Install parameter is 4 bytes long with the following byte order:

- Offset + 0 = DES features byte
- Offset + 1 = RSA features byte
- Offset + 2 = Message Digest features byte
- Offset + 3 = Signature features byte

The install parameter must be correctly defined for the features of the target card, otherwise installation will fail with a CryptoException exception.

DES features byte

- 0x00 (XXXX XX00) = No DES
- 0x01 (XXXX XX01) = Simple DES
- 0x02 (XXXX XX10) = Triple DES with 2 keys
- 0x03 (XXXX XX11) = Triple DES with 3 keys

RSA features byte

- 0x00 (XXXX XX00) = No RSA
- 0x01 (XXXX XX01) = 512 bit RSA key
- 0x02 (XXXX XX10) = 768 bit RSA key
- 0x03 (XXXX XX11) = 1024 bit RSA key

Message Digest features byte

- 0x00 (XXXX XX00) = No Message Digest
- 0x01 (XXXX XX01) = SHA-1 Message Digest
- 0x02 (XXXX XX10) = MD5 Message Digest
- 0x03 (XXXX XX11) = SHA-1 & MD5 Message Digest

Signature features byte

- 0xX0 (XXXX XX00) = No DES Signature
- 0xX1 (XXXX XX01) = DES or DES3 Signature
- 0x0X (XX00 XXXX) = No RSA Signature
- 0x1X (XX01 XXXX) = SHA-1 PKCS#1 RSA Signature (SHA-1 Message Digest MUST be set)
- 0x2X (XX10 XXXX) = MD5 PKCS#1 RSA Signature (MD5 Message Digest MUST be set)
- 0x3X (XX11 XXXX) = MD5 PKCS#1 RSA Signature and SHA-1 PKCS#1 RSA Signature
(SHA-1 Message Digest and MD5 Message Digest MUST be set)

Quick start values

Recommended install parameters according to card features are:

- Gxp211 PK or GSE in GXP211_PK card simulation mode - [02 01 01 10]
- Gxp211 PK is or GSE in GXP211_PK_IS card simulation mode - [02 01 01 10]

This Applet cannot be used on Gxp211 V2 cards.

Commands Description

DES and DES3 no padding

This command allows you to encrypt a message with DES keys. For autotest (P2=0xFF), internal keys are used. For normal mode, however, the user key must be initialized with the Put Key command (INS=E0,E1). Only ICVs of eight 0x00 bytes are supported.

INS:

0x20: DES ECB encrypt
0x22: DES ECB decrypt
0x24: DES CBC encrypt
0x26: DES CBC decrypt

P1:

0xFF: Auto test
0x00: Normal mode

P2:

0x00

Data:

P1=0xFF:
- in: None.
- out: 0x00 OK or 0x01 pattern not verified
P1=0x00:
- in: N bytes of data with N multiples of 8.
- out: N bytes of encrypted data.

DES and DES3 sign

This command allows you to encrypt a message with DES keys. For autotest (P2=0xFF), internal keys are used. For normal mode, however, the user key must be initialized with the Put Key command (INS=0xE0, 0xE1). Only ICVs of eight 0x00 bytes are supported.

INS:

0x30: DES MAC4 sign
0x36: DES MAC8 sign
0x40: DES3 MAC4 sign
0x46: DES3 MAC8 sign

P1:

0xFF: Auto test
0x00: Normal mode

P2:

0x00

Data:

P1=0xFF:
- in: None.
- out: 0x00 OK or 0x01 pattern not verified
P1=0x00:
- in : N bytes of data with N multiples of 8.
- out: 4 bytes of data for MAC4 or 8 bytes of data for MAC8.

RSA no padding

This command allows you to encrypt a message with public or private keys. For autotest (P2=0xFF), internal key are used. For normal mode, however, the user key must be initialized with the Put Key command (INS=0xF0, 0xF1, 0xF2).

INS:

0xA0: RSA encrypt (with public key)
0xA2: RSA decrypt (with private key)
0xA4: RSA decrypt (with private CRT key)

P1:
 0xFF: Auto test
 0x00: Normal mode

P2:
 0x00

Data:
 P1=*0xFF*:
 - in: None.
 - out: *0x00* OK or *0x01* pattern not verified
 P1=*0x00*:
 - in : 128 bytes of data.
 - out: 128 bytes of encrypted data.

Signature

This command allows you to sign a message with a private key. For autotest (P2=*0xFF*), internal keys are used. For normal mode, however, the user key must be initialized with the Put Key command (INS=*0xF0*, *0xF1*, *0xF2*).

INS:
 0xB0: RSA SHA1 PKCS#1 sign (with private or private CRT key)
 0xB2: RSA MD5 PKCS#1 sign (with private or private CRT key)

P1:
 0xFF: Auto test
 0x00: Normal mode
 0xNN: Update requested with NN bloc length (multiple of 0x40 for Gxp211_PK).

P2:
 0x00: Use private key
 0x01: Use private CRT key

Data:
 P1=*0xFF*:
 - in: None.
 - out: *0x00* OK or *0x01* pattern not verified
 P1=*0x00*:
 - in: 0-N bytes of data
 - out: 128 bytes of signature

Signature verification init

This command allows you to initialize the message to be verified with the Verify command.

INS:
 0xB4: RSA SHA1 PKCS#1 Verify init

P1:
 0x00

P2:
 0x00

Data:
 - In: 0-N bytes of original message for integrity verification.

Signature verification

This command allows you to verify a message's integrity by verifying its signature. The message to verify must be initialized with the Verify Init command.

INS:
 0xB6: RSA SHA1 PKCS#1 verify

P1:
 0xFF: Auto test
 0x00: Normal mode

P2:

0x00

Data:

P1=*0xFF*:

- in: none.
- out: *0x00* OK or *0x01* pattern not verified

P1=*0x00*:

- in: 128 bytes of signature to verify with the hash of the initialized message.

Message Digest

This command allows you to hash a message with SHA1 or MD5.

INS:

0xD0: SHA1 hash
0xD2: MD5 hash

P1:

0xFF: Auto test.
0x00: Normal mode.

P2:

0x00
0xNN: Update requested with NN bloc length (multiple of 0x40 for Gxp211_PK).

Data:

P1=*0xFF*:

- in: None.
- out: *0x00* OK or *0x01* pattern not verified

P1=*0x00*:

- in : 0-N bytes of data
- out: A 20 bytes hash for SHA1 or 16 bytes hash for MD5

Put DES Keys

This command allows you to initialize the user DES or DES3 keys for algorithm computation.

INS:

0xE0: Put DES key

P1:

0x00

P2:

0x00

Data:

- in: 8 bytes of data for DES key. 16 or 24 bytes of data for DES3 keys.

Put RSA keys

This command allows you to initialize the user public or private keys for algorithm computation.

INS:

0xF0: Put RSA private key
0xF4: Put RSA public key

P1:

0x00

P2:

0x00: Set the modulus.
0x01: Set the exponent.

Data:

P2=*0x00*:

- in: 64 or 96 or 128 bytes of modulus depending on key length.

P2=*0x01*:

- in: 64 or 96 or 128 bytes of exponent for private key depending on key length.
or 3-64 or 96 or 128 bytes of exponent for public key depending on key length..

Put RSA CRT key

This command allows you to initialize the user private CRT key for algorithm computation.

INS:

0xF2: Put RSA private CRT key

P1:

0x00

P2:

0x00: set P.

0x01: set Q.

0x02: set DP.

0x03: set DQ.

0x04: set PQ.

Data:

P2=*0x00*:

- in: 32 or 48 or 64 bytes of P depending on key length.

P2=*0x01*:

- in: 32 or 48 or 64 bytes of Q depending on key length.

P2=*0x02*:

- in: 32 or 48 or 64 bytes of DP depending on key length.

P2=*0x03*:

- in: 32 or 48 or 64 bytes of DQ depending on key length.

P2=*0x04*:

- in: 32 or 48 or 64 bytes of PQ depending on key length.

Generate RSA/RSA CRT key Pair

This command allows you to initialize the user public and private keys for algorithm computation.

INS:

0xF6: Generate RSA key pair

0xF8: Generate RSA CRT key pair

P1:

0x00

P2:

0x00

Data:

- in: None.

- out: None.

Get RSA keys

This command allows you to retrieve initialized user public or private keys elements.

INS:

0xFA: Get RSA private key

0xFE: Get RSA public key

P1:

0x00

P2:

0x00: get the modulus.

0x01: get the exponent.

Data:

P2=*0x00*:

- out: 1 byte of length + 64 or 96 or 128 bytes of modulus depending on key length.

P2=*0x01*:

- out: 1 byte of length + 64 or 96 or 128 bytes of exponent for private key depending on key length.
or 1 byte of length + 3 bytes of exponent for public key.

Get RSA CRT key

This command allows you to retrieve initialized user private CRT key elements.

INS:

0xFC: Get RSA private CRT key

P1:

0x00

P2:

0x00: get P.

0x01: get Q.

0x02: get DP.

0x03: get DQ.

0x04: get PQ.

Data:

P2=*0x00*:

- out: 1 byte of length + 32 or 48 or 64 bytes of P depending on key length.

P2=*0x01*:

- out: 1 byte of length + 32 or 48 or 64 bytes of Q depending on key length.

P2=*0x02*:

- out: 1 byte of length + 32 or 48 or 64 bytes of DP depending on key length.

P2=*0x03*:

- out: 1 byte of length + 32 or 48 or 64 bytes of DQ depending on key length.

P2=*0x04*:

- out: 1 byte of length + 32 or 48 or 64 bytes of PQ depending on key length.

Return error codes

6F1x = CryptoException

- 6F11 = Reason: ILLEGAL_VALUE
- 6F12 = Reason: UNINITIALIZED_KEY
- 6F13 = Reason: NO_SUCH_ALGORITHM
- 6F14 = Reason: INVALID_INIT
- 6F15 = Reason: ILLEGAL_USE

6F2x = SystemException

- 6F21 = Reason: ILLEGAL_VALUE
- 6F22 = Reason: NO_TRANSIENT_SPACE
- 6F23 = Reason: ILLEGAL_TRANSIENT
- 6F24 = Reason: ILLEGAL_AID
- 6F25 = Reason: NO_RESOURCE

6F3x = TransactionException

- 6F31 = Reason: IN_PROGRESS
- 6F32 = Reason: NOT_IN_PROGRES
- 6F33 = Reason: BUFFER_FULL
- 6F34 = Reason: INTERNAL_FAILURE

6F50 = ArrayIndexOutOfBoundsException

6F60 = IndexOutOfBoundsException

6F70 = NegativeArraySizeException

6F80 = NullPointerException

6F90 = SecurityException

6FA0 = Throwable

6FB0 = Exception

6FC0 = RuntimeException