GLOBALPLATFORM®
THE STANDARD FOR SECURE DIGITAL SERVICES AND DEVICES

GlobalPlatform Technology

Secure Channel Protocol '10'

Card Specification v2.3 – Amendment L

Version 0.0.0.8

Public Review

July 2020

Document Reference: GPC_SPE_178

THIS SPECIFICATION OR OTHER WORK PRODUCT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE COMPANY, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER DIRECTLY OR INDIRECTLY ARISING FROM THE IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT.

Contents

Figures

Tables

1 Introduction

This document specifies a secure channel protocol, named Secure Channel Protocol '10' (SCP10), based on asymmetric cryptography and a Public Key Infrastructure. SCP10 is compatible with CEN Workshop Agreement specification CWA 14890-1 ([CWA 14890-1]). It also fulfills the requirements of NICSS Framework Scheme ([NICSS-F]) defined by the Next Generation IC Card System Study Group (NICSS).

1.1 Audience

This amendment is intended primarily for card manufacturers and application developers developing GlobalPlatform card implementations.

It is assumed that the reader is familiar with smart cards and smart card production, and in particular familiar with the GlobalPlatform Card Specification [GPCS].

1.2 IPR Disclaimer

Attention is drawn to the possibility that some of the elements of this GlobalPlatform specification or other work product may be the subject of intellectual property rights (IPR) held by GlobalPlatform members or others. For additional information regarding any such IPR that have been brought to the attention of GlobalPlatform, please visit https://globalplatform.org/specifications/ip-disclaimers/. GlobalPlatform shall not be held responsible for identifying any or all such IPR, and takes no position concerning the possible existence or the evidence, validity, or scope of any such IPR.

1.3 References

The tables below list references applicable to this specification. The latest version of each reference applies unless a publication date or version is explicitly stated.

Table 1-1: Normative References

Standard / Specification	Description	Ref
GPCS	GlobalPlatform Technology Card Specification v2.3 Document Reference: GPC_SPE_034 Note: A prior definition of SCP10 occurs in Appendix F of [GPCS] v2.3 and is deprecated in v2.3.2.	[GPCS]
Cryptographic Algorithm Recommendations	GlobalPlatform Technology Cryptographic Algorithm Recommendations Document Reference: GP_TEN_053	[Crypto Rec]
ISO/IEC 7816-4	Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange	[ISO 7816-4]
ISO/IEC 7816-6	Identification cards – Integrated circuit cards with contacts – Part 6: Interindustry data elements for interchange	[ISO 7816-6]
ISO/IEC 7816-8	Identification cards – Integrated circuit cards – Part 8: Commands and mechanisms for security operations	[ISO 7816-8]

Standard / Specification	Description	Ref
ISO/IEC 7816-9	Identification cards – Integrated circuit cards – Part 9: Commands for card management	[ISO 7816-9]
ISO/IEC 7816-15	Identification cards – Integrated circuit cards – Part 15: Cryptographic information application	[ISO 7816-15]
ISO/IEC 8825-1 | ITU-T Recommendation X.690	Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)	[ISO 8825-1]
ISO/IEC 9594-8 | ITU-T Recommendation X.509	Information Technology – Open Systems Interconnection – The Directory – Public-Key and attribute certificate frameworks	[X.509]
ISO/IEC 9796-2	Information technology – Security techniques – Digital signature schemes giving message recovery – Part 2: Integer factorization based mechanisms	[ISO 9796-2]
PKCS#1	PKCS #1: RSA Cryptography Specifications v2.2	[PKCS#1]

Table 1-2: Informative References

Standard / Specification	Description	Ref
CWA 14890-1	CEN Workshop Agreement – Application Interface for smart cards used as Secure Signature Creation Devices – Part 1: Basic requirements, March 2004	[CWA 14890-1]
NICSS Framework (NICSS-F)	NICSS Prerequisites, First Edition, version 1.20, April 24, 2001, The Next generation IC Card System Study group	[NICSS-F]

1.4 Terminology and Definitions

Terms used in this document are generally defined in [GPCS].

Additional terms are listed in Table 1-3.

Table 1-3: Terminology and Definitions

Term	Definition
Application Programming Interface	A set of rules that software programs can follow to communicate with each other.
Entity Authentication	Checking the authenticity of the other party (Security Domain or Off-Card Entity) by verifying the signature of a challenge sent to the other party.
Key Authority	Authority delivering certificates (involved in a certification chain).

1.5 Abbreviations and Notations

Abbreviations and notations used in this document are included in Table 1-4.

Table 1-4: Abbreviations and Notations

Abbreviation / Notation	Meaning
APDU	Application Protocol Data Unit
API	Application Programming Interface
BER	Basic Encoding Rules
CA	Certificate Issuer
CBC	Cipher Block Chaining
CCT	Control Reference Template for Cryptographic Checksum
C-ENC	Command Encryption
CERT.KA_EX.AUT	Key Authority's Certificate(s) for External Authentication
CERT.KA_IN.AUT	Key Authority's Certificate(s) for Internal Authentication
CERT.OCE.AUT	Off-Card Entity Certificate
CERT.SD.AUT	Security Domain's Public Key
CIO	Cryptographic Information Object
CLA	CLAss byte of command message
C-MAC	Command MAC
CRT	Control Reference Template
CS	Card Secret
CT	Control Reference Template for Confidentiality
DEK	Data Encryption Session Key
ECB	Electronic Code Book
ICV	Initial Chaining Vector
INS	INStruction byte of command message
KA	Intermediate Key Authority used by CA to build certification chains
KA_OCE	Key Authority of the Off-Card Entity
Lc	Exact length of command data in a case 3 or case 4 command
Le	Maximum length of data expected in response to a case 2 or case 4 command
MAC	Message Authentication Code
N.PK.OCE.AUT	Modulus of the Off-Card Entity public key
N.PK.SD.AUT	Modulus of the Security Domain public key
NICSS	Next Generation IC Card System Study Group
OCE	Off-Card Entity

Abbreviation / Notation	Meaning
OES	Off-Card Entity Secret
P1	Reference control Parameter 1
P2	Reference control Parameter 2
PK	Public Key
PK.KA_EX.AUT	Key Authority's Public Key(s) for External Authentication
PK.OCE.AUT	Off-Card Entity's Public Key(s) for External Authentication
PK.SD.AUT	Security Domain Public Key
PK.TP_EX.AUT	Public Key of the Trust Point for External Authentication
PK.TP_IN.AUT	Public Key for Trust Point for Internal Authentication
PK.TP_OCE.AUT	Off-Card Entity Trust Point Certification Public Key
PKI	Public Key Infrastructure
R-ENCRYPTION	Response Encryption
R-MAC	Response MAC
RP / RPD	Random Padding
RSA	Rivest / Shamir / Adleman asymmetric algorithm
SCP	Secure Channel Protocol
SD	Security Domain
SIG.OCE.AUT	Signature block with the Off-Card Entity signing key
SIG.SD.AUT	Security Domain signing key
SK.OCE.AUT	Signature block with the Off-Card Entity private key
SK.SD.AUT	Security Domain private key
TLV	Tag, Length, Value
TP_EX	Trust Point for External Authentication
TP_IN	Trust Point for Internal Authentication

1.6 Revision History

GlobalPlatform technical documents numbered *n*.0 are major releases. Those numbered *n*.1, *n*.2, etc., are minor releases where changes typically introduce supplementary items that do not impact backward compatibility or interoperability of the specifications. Those numbered *n.n*.1, *n.n*.2, etc., are maintenance releases that incorporate errata and precisions; all non-trivial changes are indicated, often with revision marks.

Table 1-5: Revision History

Date	Version	Description
March 2020	0.0.0.5	Committee Review
June 2020	0.0.0.7	Member Review
July 2020	0.0.0.8	Public Review
TBD	1.0	Public Release

2 Secure Communication

Secure Channel Protocol '10' (SCP10) supports up to 19 Supplementary Logical Channels.

2.1 SCP10 Secure Channel

SCP10 is a Secure Channel Protocol based on asymmetric cryptography and Public Key infrastructure (PKI). The Secure Channel Protocol operates between a Security Domain on the card (or the Issuer Security Domain) and an Off-Card Entity (OCE) which may be the Application Provider (or Card Issuer) or another party.

2.1.1 Synchronous Mode

The synchronous implementation of SCP10 provides the following levels of security:

Authentication – in which the Security Domain authenticates the Off-Card Entity and the Off-Card Entity may authenticate the Security Domain. There are two aspects to this: Key Authentication which involves establishing mutual trust in each other's public key (PK); and Entity Authentication which involves establishing that the other party in the Secure Channel Session is the authentic owner of that public key.

Integrity and data origin authentication – in which the Security Domain and the Off-Card Entity ensure that the data being received from the other entity actually came from its claimed source in the correct sequence and has not been altered.

Confidentiality – in which confidential data is not viewable by an unauthorized entity.

A further level of security applies to specific sensitive data (e.g. cryptographic keys) which may be encrypted using a mechanism that is not part of the Secure Channel Session security.

2.1.2 Asynchronous Mode

The asynchronous mode allows off-line pre-packaging of encrypted and signed content and provides the following levels of security:

Integrity and data origin authentication – in which the Security Domain confirms the public-key-signed signature of the supplied content. This enables that the content being received from the other entity actually came from its claimed source in the correct sequence and has not been altered.

Confidentiality – in which confidential data is not viewable by an unauthorized entity. Confidential content is encrypted with the public key of the desired Security Domain.

2.2 Initiating a Secure Channel

The steps involved in initiating a Secure Channel Protocol are as follows:

1. **Certificate Verification (optional)** – The Security Domain and the Off-Card Entity (OCE) may need to traverse and verify a chain of certificates from established trust points down to each other's public key. The extent to which this is needed depends on what keys are currently validated by each party: the null condition is where they have both already validated each other's public key, in which case no explicit Certificate Verification is necessary.
2. **Entity Authentication** – The Security Domain and optionally the Off-Card Entity further check the authenticity of the other party by verifying the signature of a challenge sent to the other party.
3. **Session Key Establishment** – The two parties establish symmetric session keys for subsequent secure messaging.

Only explicit Secure Channel initiation is supported in SCP10. There is no specific command to terminate a session.

Some of the variations on the Secure Channel Protocol supported by the card or the Security Domain are announced in the parameter "i" in Card Recognition Data or Security Domain Management Data. See [GPCS] Appendix H, GlobalPlatform Data Values, for details of Card Recognition Data and Security Domain Management Data. The value "i" is coded on one byte as a bitmap as follows:

Table 2-1: Values of Parameter "i"

b8	b7	b6	b5	b4	b3	b2	b1	Description
Not available	0	0	0	0	0	-	0	Key Transport
Not available	0	0	0	0	0	-	1	Key Agreement
Not available	0	0	0	0	0	0	-	Signature with message recovery
Not available	0	0	0	0	0	1	-	Signature without message recovery

Note: "i" is a subidentifier within an object identifier, and bit b8 is reserved for use in the structure of the object identifier according to [ISO 8825-1].

Key transport and key agreement relate to the process of establishing session keys for the Secure Channel Session.

- With **key agreement**, the Security Domain and the Off-Card Entity exchange secret values when the Secure Channel is being initiated, and session keys are then derived from those secrets using an algorithm known to both the Off-Card Entity and the Security Domain.
- With **key transport**, the Security Domain receives session keys to be used for the Secure Channel Session from the Off-Card Entity during Secure Channel initiation.

Signature with message recovery and signature without message recovery refer to the signature scheme used for digital signatures in data messages during Entity Authentication. (Note that these mechanisms apply also with the signatures on digital certificates, but the value "i" does not refer to this.)

- With **signature with message recovery**, part or all of the message data that is signed is contained in the signature block and is recovered during the process of verifying the signature. Signature with message recovery is standardized in [ISO 9796-2].

- With **signature without message recovery**, the signature does not contain any part of the message data that is signed, but comprises an appendix to the complete message. Such a scheme is also known as a signature scheme ‘with appendix’. Signature without message recovery is standardized in [PKCS#1], and is also used in ITU Recommendation X.509 ([X.509]).

The Security Domain may support any combination of values of parameter “i”, but shall support at least one of the following options as defined by “i”:

- “i” = '01': Signature with recovery, key agreement
- “i” = '02': Signature without recovery, key transport

There is no requirement for a Security Domain to support more than one certificate format.

2.3 Certificate Verification

2.3.1 Overview

The Security Domain verifies a certificate of the Off-Card Entity to establish the validity of its public key. A certificate must be issued by the Trust Point for External Authentication (TP_EX) or a subordinate key authority of the TP_EX. On the other hand, the Off-Card Entity establishes the validity of the Security Domain's public key by verifying a certificate issued by the Key Authority of the Security Domain. The Security Domain shall hold a single (default) public key of the Trust Point for External Authentication (PK.TP_EX.AUT) and may also hold the validated public keys of other authorities in a certificate chain.

The Off-Card Entity may know implicitly what other public keys have already been validated by the Security Domain, and can use this knowledge to reduce the number of certificates the Security Domain is asked to verify, potentially down to zero.

By default the Security Domain expects the first certificate presented for verification to be signed by the PK.TP_EX.AUT, and each subsequent certificate to be signed by the public key validated with the previous certificate. The Off-Card Entity may request the Security Domain to use a different default initial public key by indicating it with a MANAGE SECURITY ENVIRONMENT command.

The Off-Card Entity establishes the validity of the Security Domain's public key by checking a (chain of) certificate(s).

The minimum set of keys and certificates to be stored by a Security Domain that supports asymmetric Secure Channel Protocol '10' shall be as follows:

- One Public Key for Trust Point for External Authentication (PK.TP_EX.AUT)
- One Security Domain Private Key (SK.SD.AUT)
- One Security Domain Public Key (PK.SD.AUT)
- One Security Domain Certificate (CERT.SD.AUT) corresponding to the Security Domain Public Key and signed by the Security Domain Key Authority

The Security Domain may also hold:

- The Off-Card Entity's Public Key(s) (PK.OCE.AUT) for External Authentication which has been validated
- The Key Authority's Public Key(s) (PK.KA_EX.AUT) for External Authentication in a valid certificate chain from the Trust Point for External Authentication to the Off-Card Entity
- The Key Authority's Certificate(s) for Internal Authentication (CERT.KA_IN.AUT) in a valid certificate chain from the Trust Point for Internal Authentication to the Security Domain

The minimum set of keys and certificates to be available to an Off-Card Entity (OCE) that wishes to communicate with a specific Security Domain is assumed to be as follows:

- One Public Key for Trust Point for Internal Authentication (PK.TP_IN.AUT))
- One Off-Card Entity Public Key (PK.OCE.AUT)
- One Off-Card Entity Certificate (CERT.OCE.AUT) corresponding to the Off-Card Entity Public Key and signed by the Off-Card Entity's Key Authority
- The Off-Card Entity Trust Point Certification Public Key (PK.TP_OCE.AUT) that participates in a chain of certificates down to the Security Domain's Public Key

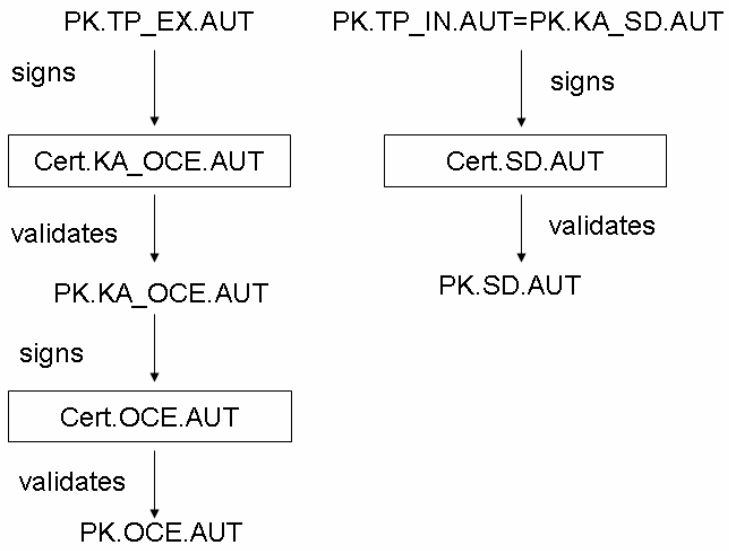
The Off-Card Entity may also hold:

- The Key Authority's Certificate(s) for External Authentication (CERT.KA_EX.AUT) in a valid certificate chain from the Trust Point for External Authentication to the Off-Card Entity.

Note that the Security Domain's owner may be considered to be the Application Provider (for a Security Domain) or the Card Issuer (for the Issuer Security Domain).

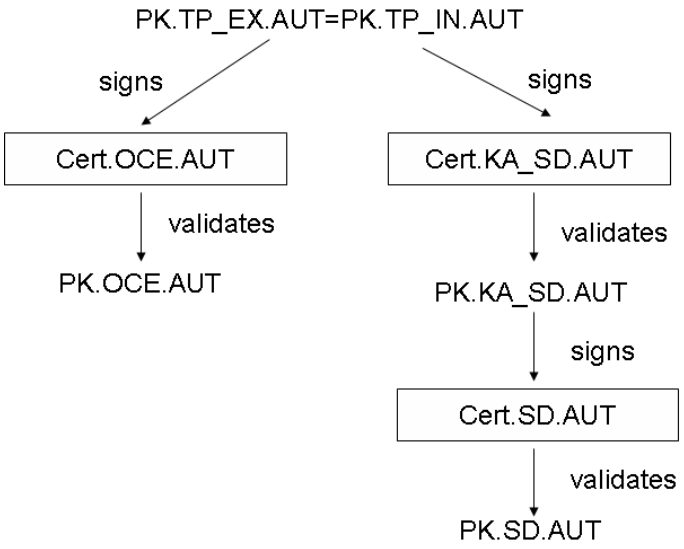
The first example in Figure 2-1 is a simple case where the Trust Point for External Authentication (TP_EX) and the Trust Point for Internal Authentication (TP_IN) certify the public key of the Key Authority of the Off-Card Entity and the Security Domain respectively, and the Key Authority of the Off-Card Entity (KA_OCE) certifies the public key of the Off-Card Entity.

Figure 2-1: Certificate Chains – Example a



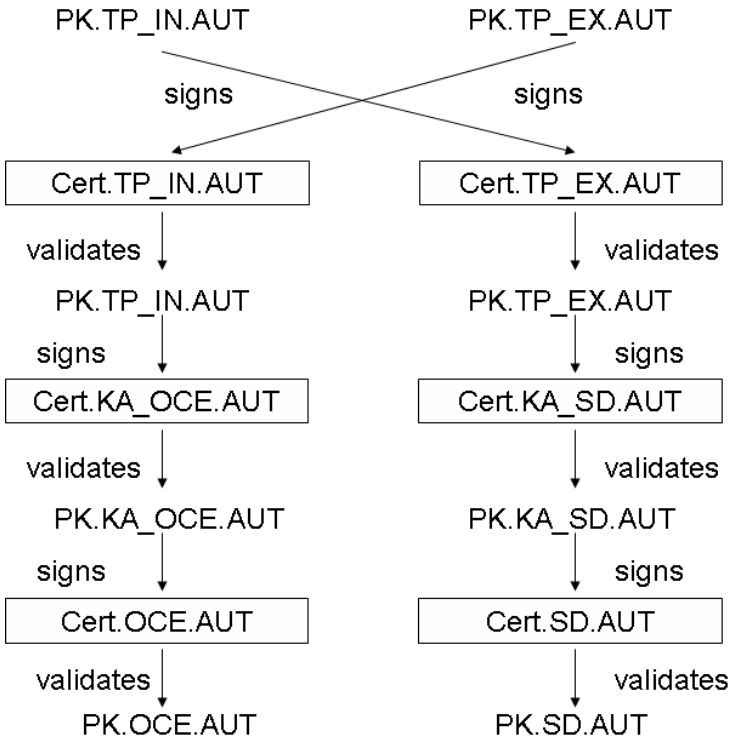
The second example in Figure 2-2 illustrates that the Trust Point for External Authentication (TP_EX) and Internal Authentication (TP_IN) directly certifies the public key of both the Off-Card Entity and the Security Domain's Key Authority; and the Security Domain's Key Authority in turn certifies the Security Domain's public key.

Figure 2-2: Certificate Chains – Example b



The third example in Figure 2-3 illustrates that the Trust Point for External Authentication (TP_EX) and Internal Authentication (TP_IN) cross-certify each other's public keys, and there are two certificate chains down to the Security Domain and OCE public keys.

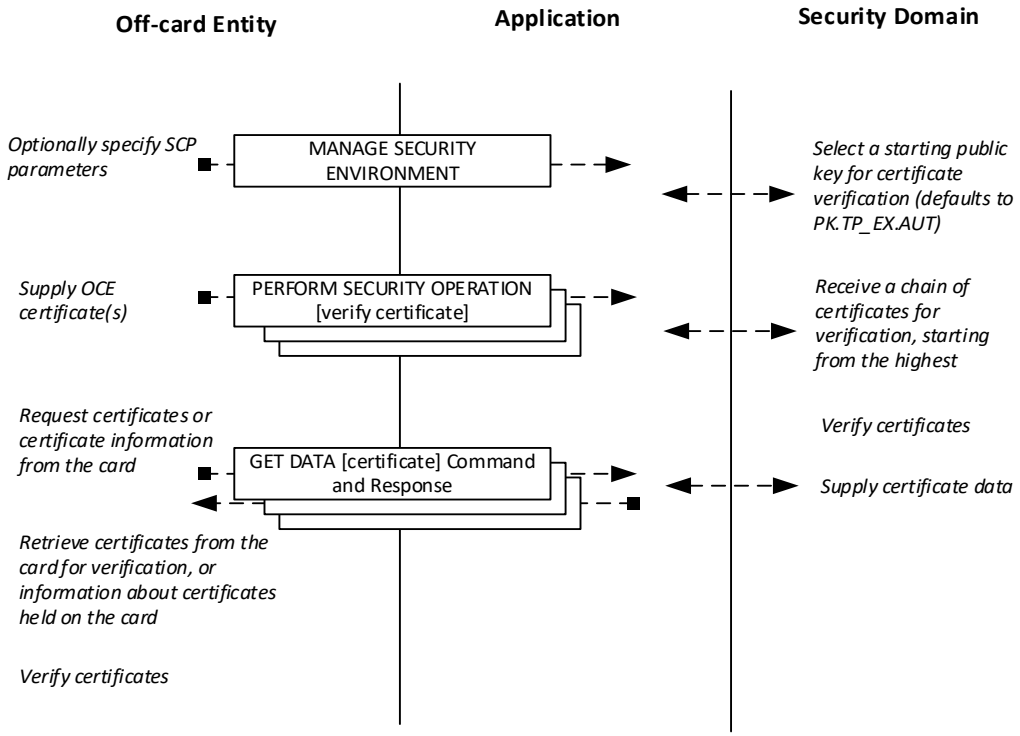
Figure 2-3: Certificate Chains – Example c



2.3.2 Certificate Verification Process Flow

The following flow is an example of the Certificate Verification stage of Secure Channel initiation, using the services of a Security Domain to perform Certificate Verification.

Figure 2-4: Certificate Verification Flow



The following commands shall be supported:

- MANAGE SECURITY ENVIRONMENT containing a ‘cryptographic mechanism reference’ designating the GlobalPlatform Secure Channel Protocol '10'. See section 5.5 for further details.
- PERFORM SECURITY OPERATION [verify certificate] command. See section 5.7 for further details.
- GET DATA [certificate] command. See section 5.3 for further details.

On receipt of a MANAGE SECURITY ENVIRONMENT command, any existing Secure Channel Session (on the same logical channel of the same card I/O interface) shall be terminated, regardless of the validity of the command: both the Current Security Level and Session Security Level are reset to NO_SECURITY_LEVEL. The MANAGE SECURITY ENVIRONMENT command may refer to a previously validated public key. If the MANAGE SECURITY ENVIRONMENT command is omitted, Security Domain default values and options shall apply, in particular for the initial default public key to use in subsequent command processing.

On receipt of a PERFORM SECURITY OPERATION [verify certificate] command not preceded by a MANAGE SECURITY ENVIRONMENT or another PERFORM SECURITY OPERATION [verify certificate] command, any existing Secure Channel Session (on the same logical channel of the same card I/O interface) shall be terminated, regardless of the result of the certificate verification: both the Current Security Level and Session Security Level are reset to NO_SECURITY_LEVEL.

Multiple PERFORM SECURITY OPERATION [verify certificate] commands may be received. In this case, a chain of certificates is presented to the Security Domain and the certificate contained in each command is verified using the Current Public Key. The Current Public Key is the public key that the Security Domain validated when verifying the last certificate presented by the Off-Card Entity during the same Secure Channel Session initiation. The Security Domain shall have a default Current Public Key at the start of a Secure Channel Session initiation phase: Trust Point for External Authentication (PK.TP_EX.AUT).

Any failure in verifying an Off-Card Entity's certificate aborts the current Secure Channel Session initiation phase, and any public keys validated during that initiation phase shall be discarded.

Multiple GET DATA [certificate] commands may be received. They may be interleaved with PERFORM SECURITY OPERATION [verify certificate] commands. The Security Domain makes no assumptions about whether the Off-Card Entity has obtained enough certificates in order to validate the Security Domain's public key. The Off-Card Entity may use the GET DATA [certificate] command for EF.OD to retrieve information about the different certificates the Security Domain holds; see section 2.3.3 for further details on EF.OD.

2.3.3 Certificate Information

The Security Domain shall support access to EF.OD with a ‘data object id’ set to '5031'. EF.OD identifies each certificate that can be retrieved by the Off-Card Entity for verification. EF.OD contains a set of Cryptographic Information Objects (CIOs) as defined in [ISO 7816-15], each of which describes a certificate and gives a pointer (in the form of a ‘data object id’) to the certificate data present within the Security Domain. The contents of EF.OD and its consistency with the certificates actually present within the Security Domain are the responsibility of the Security Domain’s Provider and beyond the control of the card.

According to [ISO 7816-15], an individual CIO for a certificate may be coded as follows (xxCertificate being of type CertificateChoice):

```
xxCertificate:
{
 commonObjectAttributes
      {
      label                Label              OPTIONAL
      flags                CommonObjectFlags OPTIONAL,
      authId               Identifier  OPTIONAL,
      userConsent          INTEGER (1..cia-ub-userConsent)        OPTIONAL,
      accessControlRules   SEQUENCE SIZE (1..MAX) OF AccessControlRule OPTIONAL,
      ….},
classAttributes
      {
      id             Identifier,
      authority      BOOLEAN      DEFAULT FALSE,
      identifier     CredentialIdentifier {{KeyIdentifiers}}        OPTIONAL,
      certHash       [0] CertHash     OPTIONAL,
      trustedUsage   [1] Usage OPTIONAL
      identifiers    [2] SEQUENCE OF CredentialIdentifier {{KeyIdentifiers}} OPTIONAL,
      validity       [4] Validity      OPTIONAL,
      ….},
 typeAttributes
      {
      value          ObjectValue {Certificate},
      ….}
}
```

Where typeAttributes vary per type of CertificateChoice, for instance x509CertificateAttributes are:

```
typeAttributes
      {
      value          ObjectValue {Certificate},
      subject        Name              OPTIONAL,
      issuer         [0] Name  OPTIONAL,
      serialNumber   CertificateSerialNumber     OPTIONAL,
      ….}
```

2.3.4 Certificate Formats

Certificate contents and encoding are outside the scope of this specification. Examples are given for illustration only. Certificates may contain various data objects and elements as defined in [ISO 7816-4], [ISO 7816-6], and [ISO 7816-8] (using application tagging), [X.509] (universal and context-specific tagging), and elsewhere. The data objects included are defined by the certificate issuer (CA).

The following table identifies some data items that appear in certificates – 'presence' indicates when the item can be expected to be present:

Table 2-2: Example of Data Included in Certificates

Name as used in [X.509]	Name as used in ISO 7816-8	Application Tag assigned by ISO 7816	Presence
Issuer	Issuer Identification Number	'42' (7816-8)	Always
Subject	Certificate holder reference OR Cardholder name	'5F20' (7816-8)	Always
SubjectPublicKey	Cardholder public key	'5F49' or '7F49' (7816-8)	Always
KeyUsage	Certificate holder authorization	'5F4C' ([ISO 7816-9])	As required
CertificateSerialNumber	Certificate serial number	-	Always
n/a	Certificate contents	'5F4E' (7816-8)	Non-self descriptive certificates
Signature	Static internal authorization OR Digital signature	'9E' (7816-8)	Always
n/a	Public key remainder	'5F38' (7816-6)	Signature scheme with recovery

Certificates may either be self-descriptive, where the signature is across individual data objects; or non-self descriptive, where the signature is across concatenated value fields, without tags and lengths. Whether the certificate to be verified by the Security Domain is self-descriptive or non-self-descriptive must be indicated in the PERFORM SECURITY OPERATION [verify certificate] command.

A Security Domain is only required to handle a single certificate format throughout a chain of certificates to be verified by the Security Domain.

Tag '67' in Card Recognition Data or Security Domain Management Data may contain one or more OIDs identifying the Security Domain's Trust Point's certification policy and/or the format of certificates that can be verified by the Security Domain and/or the format of certificates that can be retrieved from the Security Domain. They may also identify the cryptographic algorithms used for certificates, unless they are indicated in the certificates themselves.

2.3.4.1 Certificate without Message Recovery

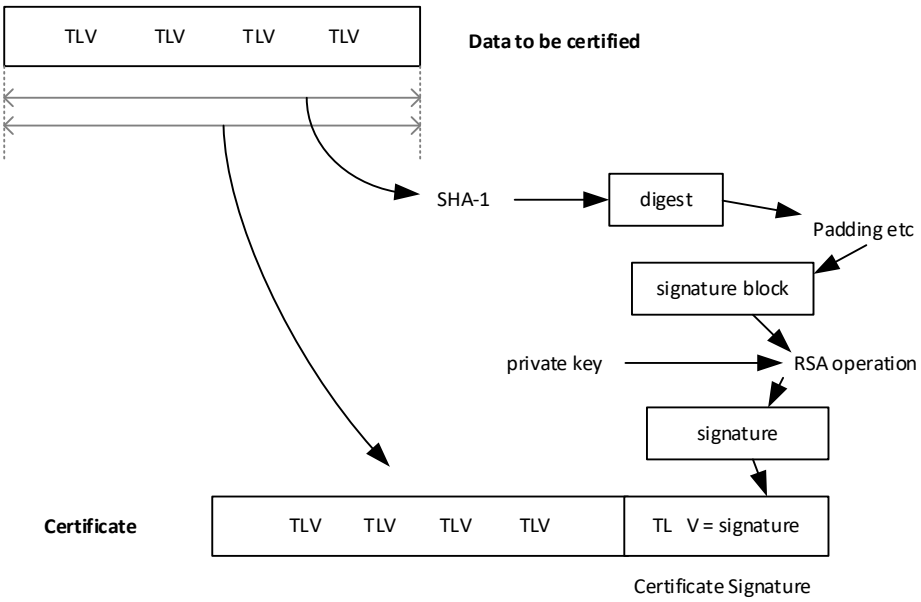
The data and public key to be certified are concatenated and a digest is created. A signature block is then prepared containing the digest and any necessary padding, constant and random data. The format of the signature block is defined by the certificate issuer (CA). The signature block size depends on the asymmetric algorithm and the certifying key size. The signature block is then signed using the certificate issuer's private certifying key, to form the value field of the Certificate Signature data object. The result is a certificate typically containing the data and public key to be certified and the Certificate Signature.

Certificate without Message Recovery, Self-Descriptive Certificate

The data objects to be certified, including the public key data object, are TLV encoded and are concatenated TLV-encoded before the digest is created and the signature block prepared. The result of the signing operation is the Certificate Signature. The certificate is TLV encoded as a series of data objects, containing all the data objects to be certified, including the public key data object and the Certificate Signature data object. X.509 certificates fall into this category.

The formation of the certificate is shown in the following example:

Figure 2-5: Certificate Formation – Self Descriptive Certificate without Message Recovery

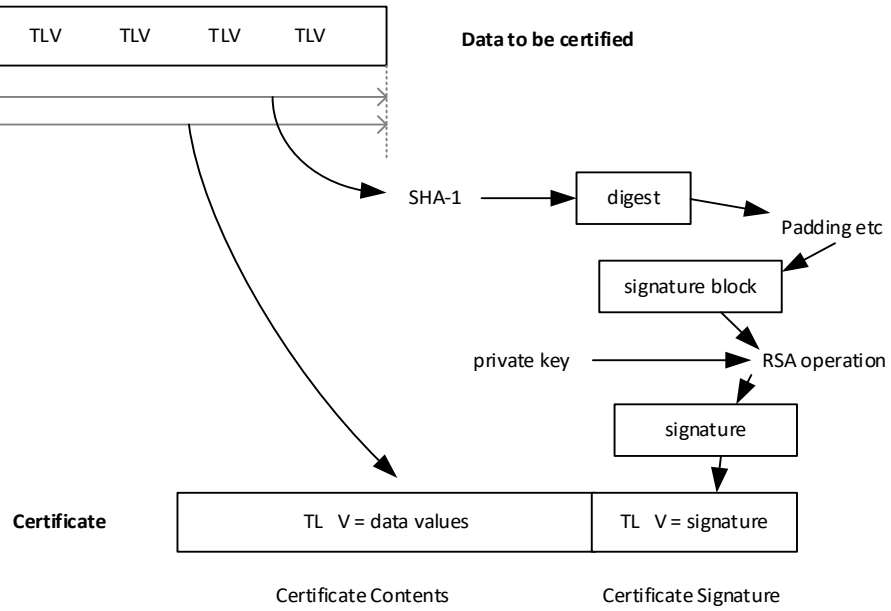


Certificate without Message Recovery, Non-Self Descriptive Certificate

The data and public key to be certified are a concatenated set of value fields. The format of the value fields included is defined by the certificate issuer (CA) and implicitly known to the recipients. A digest of the concatenated value fields is created and the signature block prepared. The result of the signing operation is the Certificate Signature. The certificate contains two data objects: the Certificate Contents, whose value is the concatenated set of value fields being certified, and the Certificate Signature.

The formation of the certificate is shown in the following example:

Figure 2-6: Certificate Formation – Non-Self Descriptive Certificate without Message Recovery



2.3.4.2 Certificate with Message Recovery

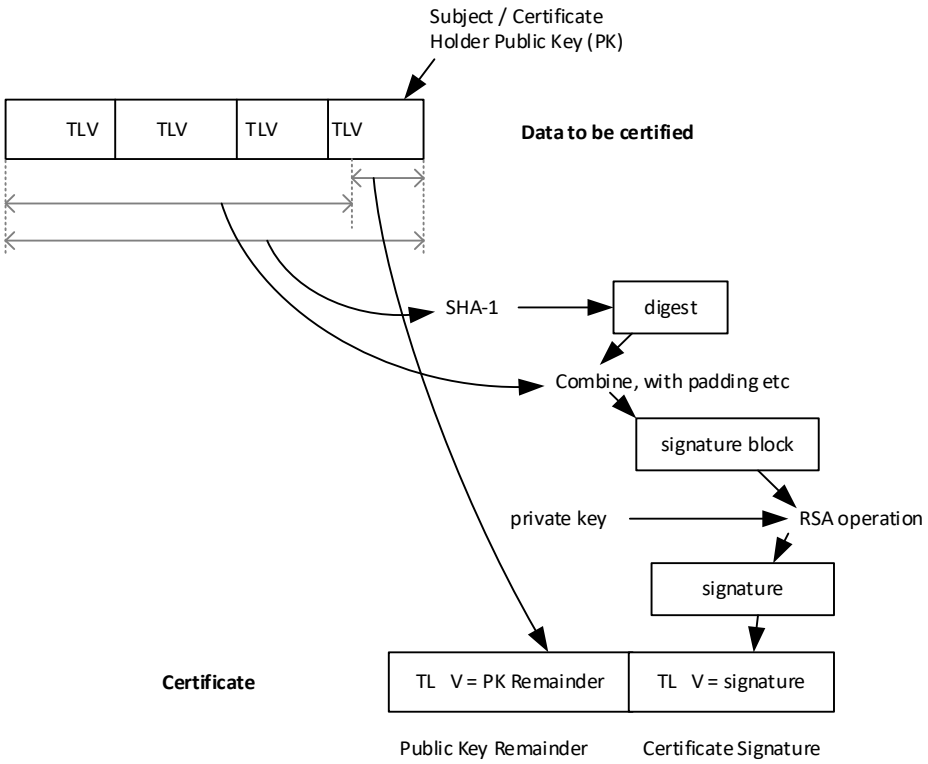
The data and public key to be certified is concatenated and a digest is created. A signature block is then prepared containing the digest, as much of the concatenated set of data and public key to be certified as can be included in the block and any necessary padding, constant and random data. The format of the signature block is defined by the certificate issuer (CA). The signature block size depends on the asymmetric algorithm and the certifying key size. The signature block is then signed using the certificate issuer's private certifying key, to form the value field of the Certificate Signature data object. Any part of the public key that could not be included in the Certificate Signature is then included in the value field of a separate Public Key Remainder data object. The result is a certificate typically containing two data objects: the Public Key Remainder and the Certificate Signature.

Certificate with Message Recovery, Self-Descriptive Certificate

The data objects to be certified, including the public key data object, are TLV encoded and are concatenated TLV-encoded before the digest is created and the signature block prepared. The result of the signing operation is the Certificate Signature.

The formation of the certificate is shown in the following diagram:

Figure 2-7: Certificate Formation – Self Descriptive Certificate with Message Recovery

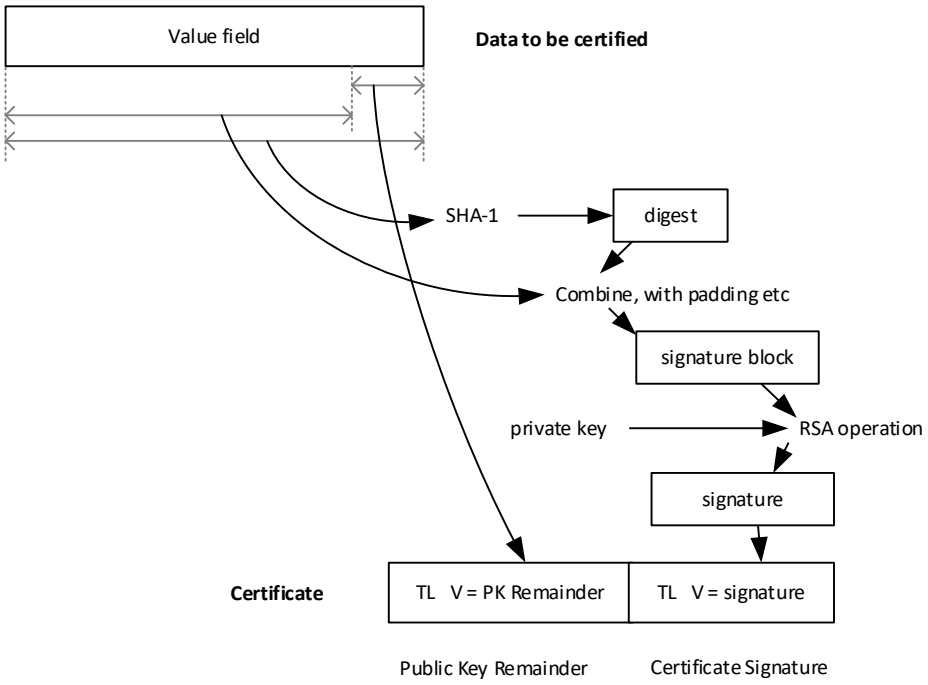


Certificate with Message Recovery, Non-Self Descriptive Certificate

The data and public key to be certified is a concatenated set of value fields. The format of the value fields included is defined by the certificate issuer (CA) and implicitly known to the recipients. A digest of the concatenated value fields is created and the signature block prepared. The result of the signing operation is the Certificate Signature.

The formation of the certificate is shown in the following diagram:

Figure 2-8: Certificate Formation – Non-Self Descriptive Certificate with Message Recovery



2.4 Entity Authentication

2.4.1 Overview of Entity Authentication

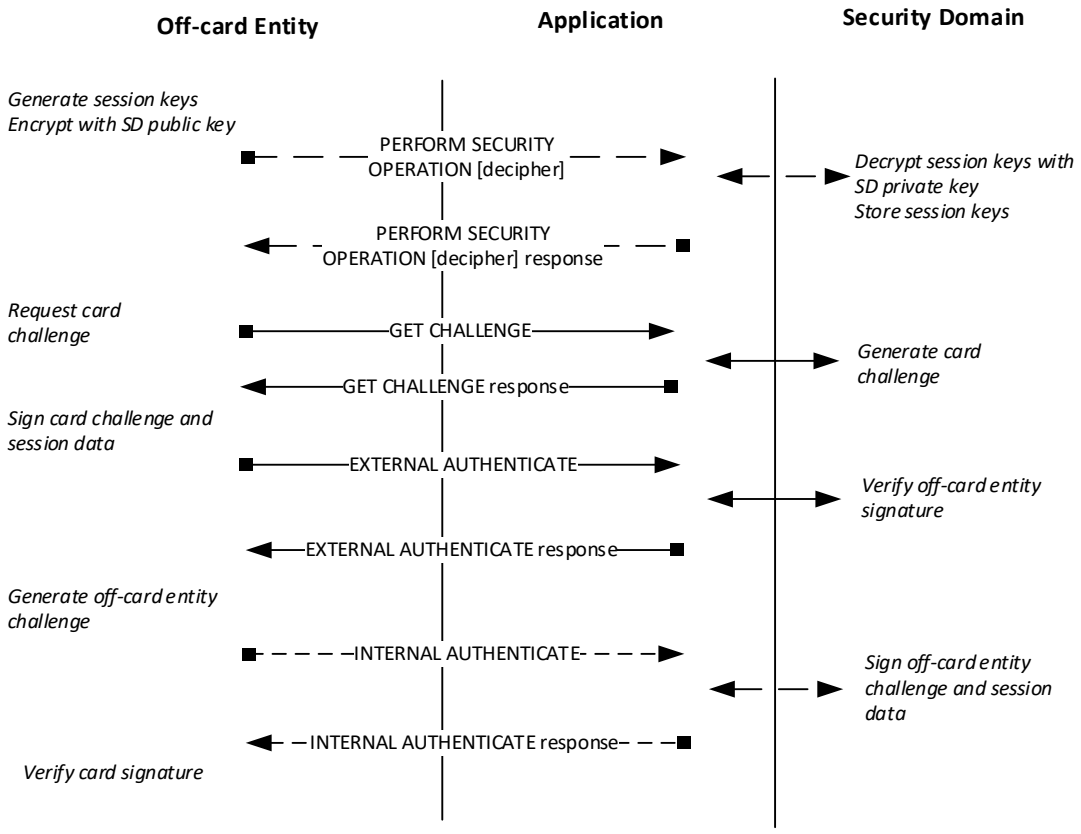
Once the parties have validated each other's public key, the Security Domain shall authenticate the Off-Card Entity using a challenge / response mechanism, and the Off-Card Entity may also authenticate the Security Domain in the same way.

Entity authentication of the Security Domain is optional, at the discretion of the Off-Card Entity. However, if the INTERNAL AUTHENTICATE command is required for use in session key agreement, then authentication of the Security Domain is performed.

2.4.2 Entity Authentication Process Flow

The following diagram gives an overview of the flow for Entity Authentication.

Figure 2-9: Entity Authentication Flow



The following commands shall be supported:

- PERFORM SECURITY OPERATION [decipher] command. See section 5.6 for further details,
- GET CHALLENGE command. See section 5.2 for further details.
- EXTERNAL AUTHENTICATE command. See section 5.1 for further details.
- INTERNAL AUTHENTICATE command. See section 5.4 for further details.

The PERFORM SECURITY OPERATION [decipher] command is optional but shall be executed when value '02' of parameter “i” is supported – see section 2.1.

The successful processing of the EXTERNAL AUTHENTICATE command (i.e. successful verification of the Off-Card Entity’s signature) shall set both the Current Security Level and Session Security Level according to the rules described in section 2.5.2, Security Level Establishment. The failed processing of the EXTERNAL AUTHENTICATE command (i.e. unsuccessful verification of the Off-Card Entity’s signature) shall reset both the Current Security Level and Session Security Level to NO_SECURITY_LEVEL and all the public keys previously verified within the current initiation phase shall be discarded.

The INTERNAL AUTHENTICATE command is optional for the key transport option and mandatory for key agreement. The INTERNAL AUTHENTICATE command shall only be received and processed once during Secure Channel Session initiation. Any error in the sequence flow or signing operation must abort the current Secure Channel Session: both the Current Security Level and Session Security Level shall be reset to NO_SECURITY_LEVEL.

2.4.3 Security Domain Authentication

The format and contents of Security Domain signature vary depending on the options chosen, as follows:

Key transport, signature without message recovery

The Security Domain signature is the result of generating a digest (hash) over a set of data, creating a signature block, and signing the signature block with the Security Domain private key (SK.SD.AUT). The data to be hashed and the contents of the signature block are as shown below.

Table 2-3: Data to Hash

Name	Length	Value	Presence
Session Key(s)	n x (16 or 24)	'xxxx...'	Mandatory
Off-Card Entity challenge	16	'xxxx...'	Mandatory

Session Keys shall be in the same order as provided in the PERFORM SECURITY OPERATION [decipher] command; see sections 4.1.2 and 5.7.

Table 2-4: Security Domain Signature Block

Name	Length	Value	Presence
Padding	2	'0001'	Mandatory
Padding ('FF')	8-n	'FF'...'FF'	Mandatory
Padding ('00')	1	'00'	Mandatory
DER encoded digest algorithm id - encoded as an object identifier	Variable	'xxxx...' (see section 3.2)	Mandatory
DER encoded Hash (length and contents depend on the digest algorithm)	Variable	'xxxx...'	Mandatory

Key agreement, signature with message recovery

The Security Domain signature is the result of generating a digest (hash) over a set of data, creating a signature block, and signing the signature block with the Security Domain private key (SK.SD.AUT). The data to be hashed and the contents of the signature block are as shown below.

Table 2-5: Data to Hash

Name	Length	Value	Presence
Random Padding (RP)	1-n	Same value as RP in Table 2-6	Mandatory
Card Secret (CS)	32	Same value as CS in Table 2-6	Mandatory
Off-Card Entity challenge	8	'xxxx...'	Mandatory
Off-Card Entity id	8	'xxxx...' (part of CERT.OCE.AUT)	Mandatory

Table 2-6: Security Domain Signature Block

Name	Length	Value	Presence
Padding	1	'6A'	Mandatory
Random Padding (RP)	1-n	Same value as RP in Table 2-5	Mandatory
Card Secret (CS)	32	Same value as CS in Table 2-5	Mandatory
Hash	20	'xxxx...'	Mandatory
Padding	1	'BC'	Mandatory

The Security Domain signature is encrypted to ensure that the Card Secret is not divulged. To do this, the minimum of the values SIG.SD.AUT and (N.PK.SD.AUT – SIG.SD.AUT) is encrypted with the Off-Card Entity public key (PK.OCE.AUT), where N.PK.SD.AUT denotes the modulus of the Security Domain public key. This ensures that the data to be encrypted is always smaller than the modulus of the Off-Card Entity public key. Note that the modulus of the Security Domain public key and modulus of the Off-Card Entity public key must have the same length in bits. See [ISO 9796-2], Digital Signature scheme 1.

2.4.4 Off-Card Entity Authentication

The format and contents of Off-Card Entity signature vary depending on the option chosen, as follows.

Key transport, signature without message recovery

The Off-Card Entity signature is the result of generating a digest (hash) over a set of data, creating a signature block, and signing the signature block with the Off-Card Entity private key (SK.OCE.AUT). The data to be hashed and the contents of the signature block are as shown below.

Table 2-7: Data to Hash

Name	Length	Value	Presence
Tag of Security Level	1	'D3'	Mandatory
Length of Security Level	1	'01'	Mandatory
Security Level	1	'xx'	Mandatory
Control reference template (CRT) tag	1	'B4' or 'B8'	Mandatory
Length of CRT	1	'00' - '7F'	Mandatory
CRT for Session Key(s)	n	'xxxx...'	Mandatory
…	…	…	…
CRT tag	1	'B4' or 'B8'	Optional
Length of CRT	1	'00' - '7F'	Conditional
CRT for Session Key(s)	n	'xxxx...'	Conditional
Card challenge	16		Mandatory

The control reference templates (CRTs) include session keys, and must be in the same order as provided in the PERFORM SECURITY OPERATION [decipher] command – see Table 5-20.

Table 2-8: Off-Card Entity Signature Block

Name	Length	Value	Presence
Padding ('0001')	2	'0001'	Mandatory
Padding ('FF')	8-n	'FF'	Mandatory
Padding ('00')	1	'00'	Mandatory
DER encoded digest algorithm id (encoded as an object identifier)	Variable	'xxxx...' (see section 3.2)	Mandatory
DER encoded hash (length and contents depend on the digest algorithm)	Variable	'xxxx...'	Mandatory

Control reference templates (CRTs) include session keys, and must be in the same order as provided in the PERFORM SECURITY OPERATION [decipher] command.

The format and contents of Card Signature vary depending on the options chosen, as discussed below.

Key agreement, signature with message recovery

The Off-Card Entity signature is the result of generating a digest (hash) over a set of data, creating a signature block, and signing the signature block with the Off-Card Entity private key (SK.OCE.AUT). The data to be hashed and the contents of the signature block are as shown below.

Table 2-9: Data to Hash

Name	Length	Value	Presence
Random Padding (RPD)	1-n	Same value as RPD in Table 2-10	Mandatory
Tag of Security Level	1	'D3'	Mandatory
Length of Security Level	1	'01'	Mandatory
Security Level	1	'xx'	Mandatory
CRT tag	1	'B4' or 'B8'	Mandatory
Length of CRT	1	'00' - '7F'	Mandatory
CRT for session key(s)	n	'xxxx...'	Mandatory
...	...	...	...
CRT tag	1	'B4' or 'B8'	Optional
Length of CRT	1	'00' - '7F'	Conditional
CRT for Session Key(s)	n	'xxxx...'	Conditional
Off-Card Entity Secret (OES)	32	Same value as OES in Table 2-10	Mandatory
Card challenge	8	'xxxx...'	Mandatory
Card id: TBD (part of CERT.SD.AUT)	8	'xxxx...'	Mandatory

Control reference templates (CRTs) exclude session keys, and must be in the same order as provided in signature block.

Table 2-10: Off-Card Entity Signature Block

Name	Length	Value	Presence
Padding	1	'6A'	Mandatory
Random Padding (RPD)	1-n	Same value as RPD in Table 2-9	Mandatory
Tag of Security Level	1	'D3'	Mandatory
Length of Security Level	1	'01'	Mandatory
Security Level	1	'xx'	Mandatory
CRT tag	1	'B4' or 'B8'	Mandatory
Length of CRT	1	'00' - '7F'	Mandatory
CRT for session key(s)	1-n	'xxxx...'	Mandatory
…	…	…	…
CRT tag	1	'B4' or 'B8'	Optional
Length of CRT	1	'00' - '7F'	Conditional
CRT for Session Key(s)	n	'xxxx...'	Conditional
Off-Card Entity Secret (OES)	32	Same value as OES in Table 2-9	Mandatory
Hash	20	'xxxx...'	Mandatory
Padding	1	'BC'	Mandatory

Control reference templates (CRTs) exclude session keys, and must be in the same order as input to the hash function.

The Off-Card Entity signature is encrypted to ensure that the Off-Card Entity secret is not divulged. To do this, the minimum of the values SIG.OCE.AUT and (N.PK.OCE.AUT – SIG.OCE.AUT) is encrypted using the Security Domain public key (PK.SD.AUT). N.PK.OCE.AUT denotes the modulus of the Off-Card Entity public key. This ensures that the data to be encrypted is always smaller than the modulus of the Security Domain public key. Note that the modulus of the Security Domain public key and modulus of the Off-Card Entity public key must have the same length in bits. See [ISO 9796-2], Digital Signature scheme 1.

2.5 Session Key and Security Level Establishment

When using the key transport option, Entity Authentication is preceded by the session keys and requested Security Level being sent to the Security Domain using the PERFORM SECURITY OPERATION [decipher] command; the Security Domain stores them until session initiation is complete.

A Security Domain supporting the key transport option shall decrypt with its private key (SK.SD.AUT or if the Security Domain has more than one key pair, the private key identified in the MANAGE SECURITY ENVIRONMENT command) the command data field of the PERFORM SECURITY OPERATION [decipher] command. Any failure in the decryption operation aborts the current Secure Channel Session initiation phase, and any public keys validated during that initiation phase shall be discarded.

In the key transport option the Secure Channel Session is established after successful processing of the EXTERNAL AUTHENTICATE command. An INTERNAL AUTHENTICATE command can be issued immediately after the EXTERNAL AUTHENTICATE command without secure messaging.

In the key agreement option the Secure Channel Session is established after successful processing of the EXTERNAL AUTHENTICATE and INTERNAL AUTHENTICATE commands.

2.5.1 Session Key Establishment

The Off-Card Entity supplies the Security Domain with details of what session keys are to be established. This information is in the form of ‘control reference templates’ (see section 4.1.2) which are supplied either in the PERFORM SECURITY OPERATION [decipher] command (with the key transport option) or in the EXTERNAL AUTHENTICATE command (with the key agreement option).

If the key transport option is used, then the session keys are provided by the Off-Card Entity within the ‘control reference templates’ (see section 4.1.2).

If the key agreement option is used, then the secrets exchanged between the Off-Card Entity and the Security Domain during the Entity Authentication process are used to establish session keys, as defined in section 4.1.

Once session keys have been established successfully, ICV sequence counter(s), used for secure messaging on subsequent commands and responses, are initialized as described in section 4.2, Secure Messaging.

2.5.2 Security Level Establishment

The requested Security Level is supplied in the PERFORM SECURITY OPERATION [decipher] command (with the key transport option) or in the EXTERNAL AUTHENTICATE command (with the key agreement option).

The successful initiation of a Secure Channel Session shall set the Current Security Level and Session Security Level to the requested Security Level combined with the AUTHENTICATED or ANY_AUTHENTICATED indicator (see [GPCS] section 10.4.2, Authentication with Asymmetric Cryptography, for further details). If the requested Security Level is set to zero, the successful initiation of a Secure Channel Session shall set the Current Security Level and Session Security Level to AUTHENTICATED or ANY_AUTHENTICATED only.

2.6 Protocol Rules

The Current Security Level of a communication not included in a Secure Channel Session shall be set to NO_SECURITY_LEVEL. In accordance with the general rules described in [GPCS] Chapter 10, Secure Communication, the following rules shall apply:

- The successful initiation of a Secure Channel Session shall set the Current Security Level to the requested Security Level from the selected Application's perspective: it is at least set to AUTHENTICATED or ANY_AUTHENTICATED (see [GPCS] section 10.4.2, Authentication with Asymmetric Cryptography, for details.
- The Current Security Level shall apply to the entire Secure Channel Session unless successfully modified at the request of the Application.
- When the Current Security Level is set to NO_SECURITY_LEVEL, then:
  - If the Secure Channel Session was aborted during the same Application Session, the incoming command shall be rejected with a security error.
  - Otherwise no security verification of the incoming command shall be performed. The Application processing the command is responsible for applying its own security rules.
- If a Secure Channel Session is active for incoming commands (i.e. Current Security Level at least set to either AUTHENTICATED or ANY_AUTHENTICATED), the security of the incoming command shall be checked according to the Current Security Level, or if the APDU class byte indicates Secure Messaging and Secure Messaging data objects are present in the command data field:
  - When the security of the command does not match or exceed the Current Security Level, the command shall be rejected with a security error, the Secure Channel Session aborted and the Current Security Level reset to NO_SECURITY_LEVEL.
  - If a security error is found, the command shall be rejected with a security error, the Secure Channel Session aborted and the Current Security Level reset to NO_SECURITY_LEVEL.
  - If (one of) the appropriate session key(s) is not available, the command shall be rejected with a security error, the Secure Channel Session aborted and the Current Security Level reset to NO_SECURITY_LEVEL.
  - In all other cases, the Secure Channel Session shall remain active and the Current Security Level shall reflect the level of security established by the current command (e.g. C-MAC and/or C-ENCRYPTION). The Application is responsible for further processing the command.
- If a Secure Channel Session is active for outgoing responses (i.e. Current Security Level at least set to AUTHENTICATED or ANY_AUTHENTICATED), secure messaging protection shall be applied to the outgoing response according to the Current Security Level (i.e. R-MAC and/or R-ENCRYPTION):
  - If a cryptographic error occurs, a security error shall be returned, the Secure Channel Session aborted and the Current Security Level reset to NO_SECURITY_LEVEL.
  - If (one of) the appropriate session key(s) is not available, a security error shall be returned, the Secure Channel Session aborted and the Current Security Level reset to NO_SECURITY_LEVEL.
  - Otherwise, the Secure Channel Session shall remain active and the Current Security Level unmodified.
- If a Secure Channel Session is aborted, it is still considered not terminated.
- If the Security Domain supports application data encryption and/or decryption, it shall decrypt or encrypt a block of secret data upon request. If the service is not supported or if (one of) the appropriate cryptographic key(s) is not available, the request shall be rejected but the Current Security Level, Session Security Level and Secure Channel Session in operation (if any) shall not be impacted.

- The current Secure Channel Session shall be terminated (if aborted or still open), both the Current Security Level and Session Security Level reset to NO_SECURITY_LEVEL on either:
  - Attempt to initiate a new Secure Channel Session
  - Termination of the Application Session (e.g. new Application selection)
  - Termination of the associated logical channel
  - Termination of the Card Session (card reset or power off)
  - Explicit termination by the Application (e.g. invoking GlobalPlatform API)

3 Cryptographic Algorithms

The cryptographic and hashing algorithms described in [GPCS] Appendix B, Algorithms (Cryptographic and Hashing), apply to SCP10. This section defines the additional requirements for SCP10.

3.1 Asymmetric Cryptography

In this specification, signing means the deciphering of a signature block using the signer's private RSA key. The RSA key shall have a minimum length of 2048; for additional recommendations, see GlobalPlatform Cryptographic Algorithm Recommendations ([Crypto Rec]).

For certificates to be verified by the card, and message signatures signed and verified by the card, the cryptographic scheme shall be RSA. The signature block and signature are the same length as the key modulus.

In Entity Authentication, Digital Signature Scheme 1 in [ISO 9796-2] shall be used for signature with message recovery, and the signature scheme with appendix RSASSA-PKCS1-V1_5 in [PKCS#1] for signature without message recovery.

The details of the signature scheme for certificates shall be either implicitly known by the Off-Card Entity or specified by the Security Domain Trust Point through the contents of Card Recognition Data or Security Domain Management Data in tag '67'.

In the key transport option, the cryptographic scheme for encrypting the session keys and their CRT templates shall be RSA according to the encryption scheme RSA-OAEP as defined in [PKCS#1].

3.2 Digest Algorithm

The default digest algorithm for use in conjunction with SCP10 asymmetric cryptography in Entity Authentication shall be SHA-256 for this version of the Specification. An alternative algorithm may be specified in Card Recognition Data or Security Domain Management Data in tag '67'.

The Object Identifier for SHA-256 is:

```
{joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3)
nistAlgorithm(4) hashAlgs(2) sha256(1)}
```

which is DER-TLV encoded as '60 86 48 01 65 03 04 02 01'.

3.3 Message Integrity ICV

The ICV for each C-MAC and R-MAC calculation is obtained by enciphering an ICV sequence counter. The ICV sequence counter is initialized during Secure Channel initiation to either:

- the value supplied in tag '91' of the control reference template, for the key transport option (separate initial value for each key), or
- the concatenation of the last 4 bytes of the card secret and the last 4 bytes of Off-Card Entity secret, for the key agreement option (same initial value for all keys).

The ICV sequence counter is incremented by one for each C-MAC and R-MAC. For calculating a C-MAC ICV, the ICV sequence counter is AES enciphered using the first part of the Secure Channel C-MAC key. For calculating an R-MAC ICV, the ICV sequence counter is AES enciphered using the first part of the Secure Channel R-MAC key.

3.4 Message Integrity C-MAC and R-MAC

Message integrity is achieved by applying a MAC to message data. The MAC may be:

- C-MAC for APDU command messages (generated by the Off-Card Entity)
- R-MAC for APDU response messages (generated by the card)

The receiving entity, on receipt of the message containing a MAC, using the same session key, performs the same operation and by comparing its generated MAC with the MAC received from the sending entity is assured of the integrity of the full command or response.

The integrity of the sequence of APDU command or response messages being transmitted to the receiving entity is achieved by using an encrypted sequence counter as part of the MAC generation. This ensures the receiving entity that all messages in a sequence have been received.

3.5 APDU Encryption and Decryption for Message Confidentiality

Message confidentiality is achieved by encrypting the whole of the command or response data field. This includes any data within the data field that has already been protected for another purpose, such as secret or private keys encrypted with the data encryption key.

4 Cryptographic Usage

4.1 AES Session Keys

4.1.1 Overview

All session keys shall be AES keys.

The Off-Card Entity supplies information on cryptographic keys to be established for the session in a set of control reference templates. Each control reference template specifies the usage of the key.

In the case of key transport, the templates are supplied in the PERFORM SECURITY OPERATION [decipher] command and contain the actual key values.

In the case of key agreement, the templates are supplied in the EXTERNAL AUTHENTICATE command, and do not contain the key values.

4.1.2 Control Reference Templates

A control reference template (CRT) is structured as follows:

Table 4-1: Single CRT

Tag	Length	Name	Presence
'B4' or 'B8'	'00' - '7F'	CRT tag = 'B4' (CCT) or 'B8' (CT)	Mandatory
'95'	1	Key Usage Qualifier = '10' – secure messaging for commands '20' – secure messaging for responses '30' – secure messaging for commands and responses '40' – encipherment of sensitive data in responses '80' – encipherment of sensitive data in commands 'C0' – encipherment of sensitive data in commands & responses	Mandatory
'80'	0 or 1	Optional cryptographic mechanism. Contains Key Type, coded according to [GPCS] Table 11-16, Key Type Coding	Optional
'D1'	0, 16, or 24	Off-Card Entity Session Key	Conditional
'91'	0 or 8	Initial value of sequence counter, for use in secure messaging	Conditional

Note: Key Usage Qualifier and the CRT tag together define the key usage, which is C-MAC and/or R-MAC for control reference template 'B4', and C-ENC and/or R-ENC or DEK for control reference template 'B8'.

The Key Type identifies the cryptographic algorithm.

The Off-Card Entity Session Key and sequence counter data objects are only present with the key transport option.

The sequence counter data object is used in secure messaging as the initial value for the ICV sequence counter for this key, which is encrypted as defined in section 3.3, Message Integrity ICV, to derive the ICV for the first MAC generated using this key.

4.1.3 Session Key Derivation

With the key agreement option, session keys are derived from the secret data exchanged during Secure Channel initiation as follows:

- The two 32-byte secrets Off-Card Entity Secret and Card Secret are exclusive or-ed, giving result (1).
- A 32-byte binary counter is set to a value depending on the key usage and its position in the set of CRTs supplied, as shown below.
- Result (1) is appended with a 32 bit binary counter with the appropriate value for this key, and the result is hashed using SHA-1, giving result (2).
- Bytes 1-16 of result (2) form the AES session key.

Table 4-2: Counter Value for Session Key Calculation

Counter Value	Key
1	The MAC key whose CRT is first in the set of CRTs supplied by the Off-Card Entity
2	The ENC key whose CRT is first in the set of CRTs supplied by the Off-Card Entity
3	A subsequent MAC key, if any
4	A subsequent ENC key, if any
5	The data encryption key whose CRT is first in the set of CRTs supplied by the Off-Card Entity
6	A subsequent data encryption key, if any

4.2 Secure Messaging

4.2.1 APDU Command C-MAC Protection

This section applies where command integrity (C-MAC) is required but not command confidentiality (C-ENC).

A C-MAC is generated by an Off-Card Entity and applied across the full APDU command being transmitted to the card including the header, the command data field (if present) and Le (if present). Input data to the MAC calculation is first prepared as defined in [ISO 7816-4]:

- The following data is concatenated:
  - The command header CLA, INS, P1, P2 from the unprotected APDU, with the logical channel bits in the CLA byte set to zero, and appended with four bytes '80 00 00 00'
  - If a command data field is present in the unprotected APDU, a BER-TLV data object with tag '81' containing the complete original command data field, regardless of its contents and format
  - If Le is present in the unprotected APDU, a BER-TLV data object with tag '97' containing the original Le value
- AES padding is applied as defined in [GPCS] section B.1.3.

A C-MAC is generated using the Secure Channel C-MAC session key, the encrypted sequence counter as the ICV as defined in section 3.3, and the signature method described in [GPCS] section B.2.2, AES Plus MAC, across the input data.

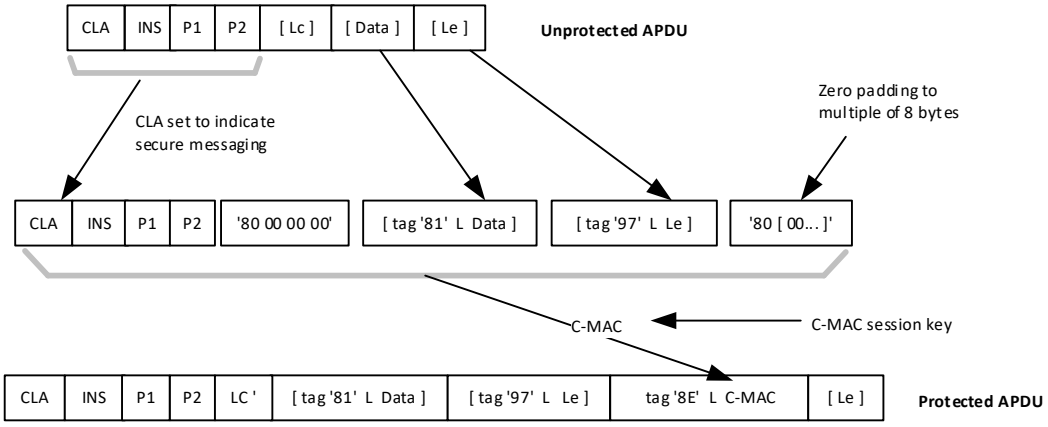
To reflect the presence of a C-MAC in the command message, the unprotected APDU shall be modified as follows:

- The class byte shall be modified to indicate that this APDU command includes secure messaging. This is achieved by setting to ‘11’ bits 4-3 of a class byte indicating a logical channel number 0 to 4 (unprotected CLA set to '00' - '03' or '80' - '83') or by setting to ‘1’ bit b6 of a class byte indicating a logical channel number 4 to 19 (unprotected CLA set to '40' - '4F' or 'C0' - 'CF'); see [GPCS] section 11.1.4. The logical channel bits are unchanged.
- The length of the command message (Lc) shall be incremented by:
  - 10 bytes to allow for the C-MAC data object, plus
  - 2 or more bytes to allow for the tag and length of the command data field data object (if command data is present - note: the length field may be longer than one byte), plus
  - 3 bytes to allow for the Le data object (if Le is present).
- The command data, if present in the unprotected APDU, shall be encapsulated in a BER-TLV data object with tag '81' and a length field coded according to [ISO 8825-1].
- The Le byte, if present in the unprotected APDU, shall be contained in a BER-TLV data object with tag '97'.
- The C-MAC shall be encapsulated in a BER-TLV data object with tag '8E' and appended at the end of the command data field.

No C-MAC padding is present in the transmitted APDU.

The following diagram shows the message reformatting that is performed by the Off-Card Entity when a command is protected for integrity.

Figure 4-1: Secure Messaging: Command Message Protected for Integrity



The card, in order to verify the C-MAC, shall perform the same procedure as employed by the Off-Card Entity in order to verify the C-MAC. The ICV sequence counter used in ICV calculation is then incremented. This is true regardless of whether the APDU processing completes successfully or not; i.e. a new sequence counter value shall always be used for the next C-MAC or R-MAC.

4.2.2 APDU Command C-ENC Protection

This section applies where command confidentiality (C-ENC) is required but not command integrity (C-MAC).

No encryption shall be applied to a command where there is no command data field: in this case the command message (header and optional Le) is sent without modification.

Otherwise the Off-Card Entity encrypts the command data field of the command message being transmitted to the card. This includes any data within the data field that has already been protected for another purpose; e.g. secret or private keys encrypted with the data encryption session key.

Prior to encrypting the data, AES padding is applied as defined in [GPCS] section B.2.3.

The padded command data field is enciphered using AES in CBC mode as defined in [GPCS] section B.2.1, the C-ENC session key established during the Secure Channel initiation process.

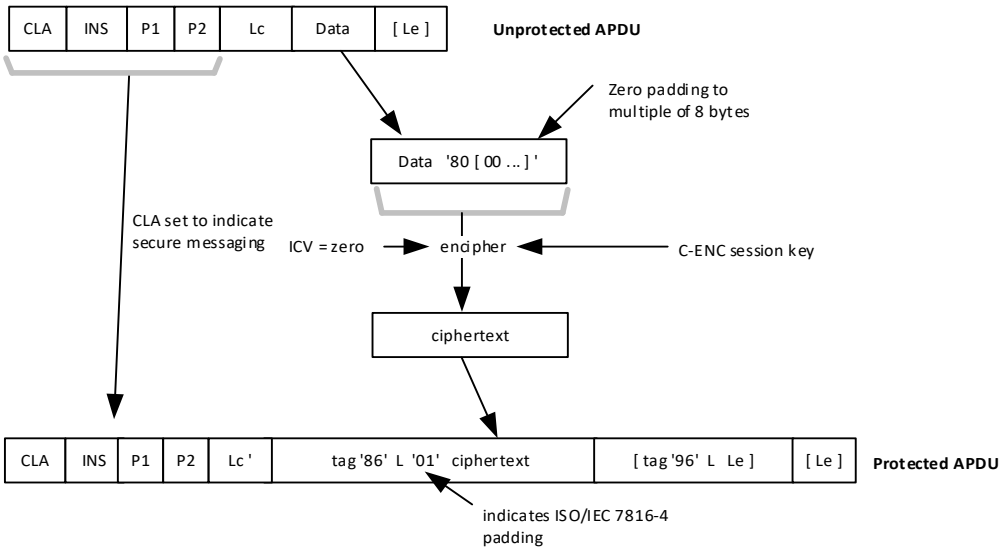
To reflect the C-ENC protection of the command, the unprotected APDU shall be modified as follows:

- The class byte shall be modified to indicate that this APDU command includes secure messaging. This is achieved by setting to ‘10’ bits 4-3 of a class byte indicating a logical channel number 0 to 4 (unprotected CLA set to '00' - '03' or '80' - '83') or by setting to ‘1’ bit b6 of a class byte indicating a logical channel number 4 to 19 (unprotected CLA set to '40' - '4F' or 'C0' - 'CF'); see [GPCS] section 11.1.4. The logical channel bits are unchanged.
- Lc shall be incremented by:
  - 4 or more bytes to allow for the tag and length of the command data field data object, the padding indicator and the variable padding (the length field may be longer than one byte), plus
  - 3 bytes to allow for the Le data object (if Le is present).

- The encrypted command data shall be preceded by the ISO/IEC 7816 padding indicator '01' and encapsulated in a BER-TLV data object with tag '86' and a length field coded according to [ISO 8825-1].
- The Le byte, if present in the unprotected APDU, shall be contained in a BER-TLV data object with tag '96'.

The following diagram shows the message reformatting that is performed by the Off-Card Entity when a command is protected for confidentiality.

Figure 4-2: Secure Messaging: Command Message Protected for Confidentiality



4.2.3 APDU Command C-MAC and C-ENC Protection

This section applies where both command confidentiality (C-ENC) and integrity (C-MAC) are required.

No encryption shall be applied to a command where there is no command data field: in this case the message shall be protected as defined in section 4.2.1, APDU Command C-MAC Protection.

Otherwise the Off-Card Entity first encrypts the command data field of the command message being transmitted to the card as defined in section 4.2.2.

A C-MAC is generated by an Off-Card Entity as defined in section 4.2.1. Input data to the MAC calculation is first prepared as defined in [ISO 7816-4]:

- The following data is concatenated:
  - The command header CLA, INS, P1, P2 from the unprotected APDU, with the logical channel bits in the CLA byte set to zero, appended with four bytes '80 00 00 00'
  - If a command data field is present in the unprotected APDU, a BER-TLV data object with tag '87' containing the ISO/IEC 7816 padding indicator '01' followed by the encrypted command data
  - If Le is present in the unprotected APDU, a BER-TLV data object with tag '97' containing the original Le value
- AES padding is applied as defined in [GPCS] section B.1.3.

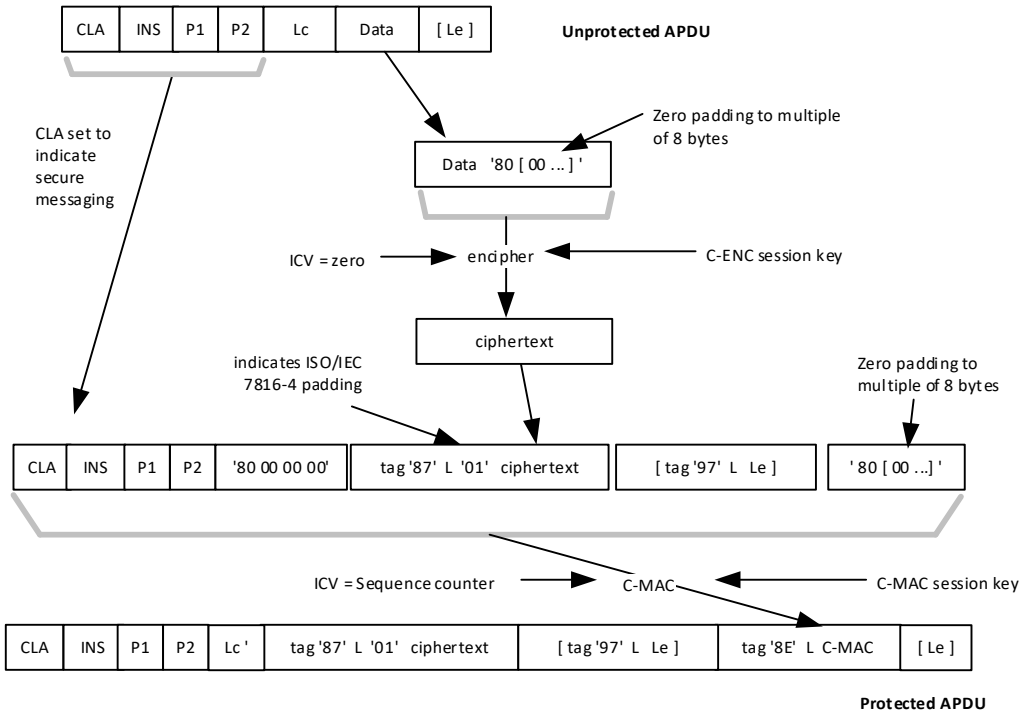
To reflect the presence of a C-MAC and C-ENC protection of the command, the unprotected APDU shall be modified as follows:

- The class byte shall be modified to indicate that this APDU command includes secure messaging. This is achieved by setting to '11' bits 4-3 of a class byte indicating a logical channel number 0 to 4 (unprotected CLA set to '00' - '03' or '80' - '83') or by setting to '1' bit b6 of a class byte indicating a logical channel number 4 to 19 (unprotected CLA set to '40' - '4F' or 'C0' - 'CF'); see [GPCS] section 11.1.4. The logical channel bits are unchanged.
- Lc shall be incremented by:
  - 10 bytes to allow for the C-MAC data object, plus
  - 4 or more bytes to allow for the tag and length of the command data field data object, the padding indicator and the variable padding (the length field may be longer than one byte), plus
  - 3 bytes to allow for the Le data object (if Le is present).
- The encrypted command data shall be preceded by the ISO/IEC 7816 padding indicator '01' and encapsulated in a BER-TLV data object with tag '87' and a length field coded according to [ISO 8825-1].
- The Le byte, if present in the unprotected APDU, shall be contained in a BER-TLV data object with tag '97'.
- The C-MAC shall be encapsulated in a BER-TLV data object with tag '8E' and appended at the end of the command data field.

No C-MAC padding is present in the transmitted APDU.

The following diagram shows the message reformatting that is performed by the Off-Card Entity when a command is protected for integrity and confidentiality.

Figure 4-3: Secure Messaging: Command Message Protected for Integrity and Confidentiality



4.2.4 APDU Response R-MAC Protection

This section applies where response integrity (R-MAC) is required but not confidentiality (R-ENC).

No R-MAC shall be generated and no protection shall be applied to a response where status bytes SW1 and SW2 indicate an error: in this case only status bytes shall be returned in the response.

When R-MAC protection is required for a case 1 or case 3 command, the card shall process the command as a case 2 or case 4 command respectively and treat Le as if it were present and set to zero.

An R-MAC is generated by the card across the response data field (if present) and status bytes. Input data to the MAC calculation is first prepared as defined in [ISO 7816-4]:

- The following data is concatenated:
  - If a response data field is present in the unprotected APDU, a BER-TLV data object with tag '81' containing the complete original response data field, regardless of its contents and format
  - A BER-TLV data object with tag '99', containing the original status word SW1 - SW2 value
- AES padding is applied as defined in [GPCS] section B.2.3.

An R-MAC is generated using the Secure Channel R-MAC session key, the encrypted sequence counter as the ICV as defined in section 3.3, and the signature method described in [GPCS] section B.2.2, AES MAC, across the input data.

To reflect the presence of an R-MAC protection of the response, the unprotected APDU shall be modified as follows:

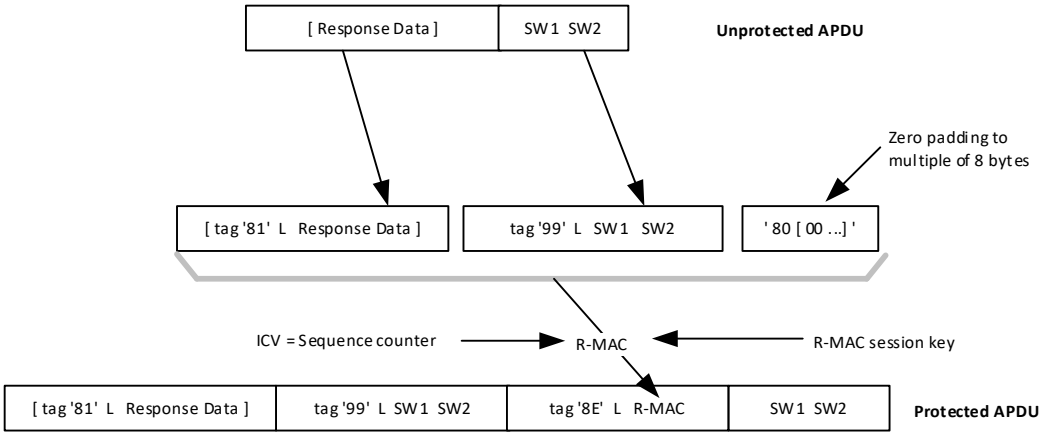
- The response data, if present in the unprotected APDU, shall be encapsulated in a BER-TLV data object with tag '81' and a length field coded according to [ISO 8825-1].
- The status word SW1 - SW2 of the unprotected APDU shall be contained in a BER-TLV data object with tag '99'.
- The R-MAC shall be encapsulated in a BER-TLV data object with tag '8E' and appended at the end of the response data field.

No R-MAC padding is present in the transmitted APDU.

The Off-Card Entity, in order to verify the R-MAC, shall perform the same processing in order to generate an R-MAC and compare it with the transmitted R-MAC.

The following diagram shows the message reformatting that is performed by the card when a response message is protected for integrity.

Figure 4-4: Secure Messaging: Response Message Protected for Integrity



4.2.5 APDU Response R-ENC Protection

This section applies where response confidentiality (R-ENC) is required but not integrity (R-MAC).

No protection shall be applied to a response where status bytes SW1 and SW2 indicate an error or where there is no response data field: in this case only status bytes shall be returned in the response.

Otherwise, the Security Domain encrypts the response data field. This includes any data within the data field that has already been protected for another purpose, such as secret or private keys encrypted with the data encryption session key.

Prior to encrypting the response data field, AES padding is applied as defined in [GPCS] section B.2.3.

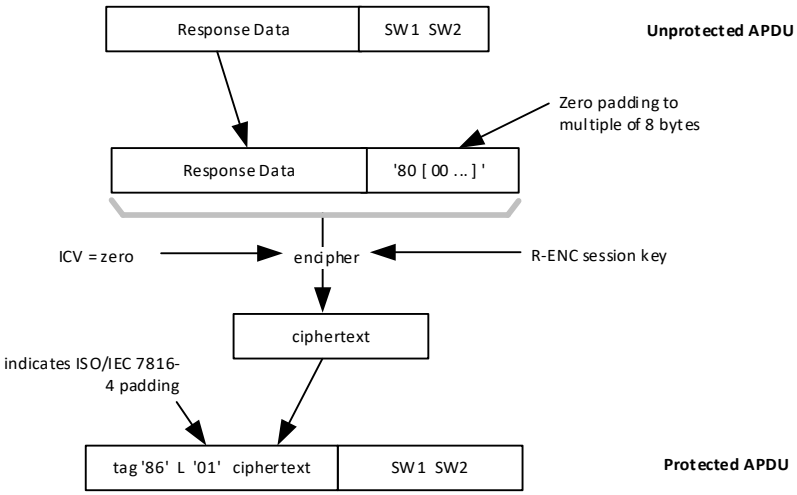
The padded response data field is then enciphered using AES in CBC mode as defined in [GPCS] section B.2.1, CBC Mode, the R-ENC session key established during the Secure Channel initiation process.

To reflect the R-ENC protection of the response, the unprotected APDU shall be modified as follows:

- The encrypted response data shall be preceded by the ISO/IEC 7816 padding indicator '01' and encapsulated in a BER-TLV data object with tag '86' and a length field coded according to [ISO 8825-1].

The following diagram shows the message reformatting that is performed by the card when a response message is protected for confidentiality.

Figure 4-5: Secure Messaging: Response Message Protected for Confidentiality



4.2.6 APDU Response R-MAC and R-ENC Protection

This section applies where both response confidentiality (R-ENC) and response integrity (R-MAC) are required.

No R-MAC or encryption shall be applied to a response where status bytes SW1 and SW2 indicate an error: in this case only status bytes shall be returned in the response.

No encryption shall be applied to a response where there is no response data field: in this case the message shall be protected as defined in section 4.2.4, APDU Response R-MAC Protection.

Otherwise, the card first encrypts the response data field of the response message being transmitted to the Off-Card Entity as defined in section 4.2.5.

An R-MAC is then generated by the card as defined in section 4.2.4. Input data to the MAC calculation is first prepared as defined in [ISO 7816-4]:

- The following data is concatenated:
  - If a response data field is present in the unprotected APDU, a BER-TLV data object with tag '87' containing the ISO/IEC 7816 padding indicator '01' followed by the encrypted response data
  - A BER-TLV data object with tag '99' containing the original SW1 - SW2 status word value
- AES padding is applied as defined in [GPCS] section B.2.3.

To reflect the presence of an R-MAC and R-ENC protection of the response, the unprotected APDU shall be modified as follows:

- The encrypted response data shall be preceded by the ISO/IEC 7816 padding indicator '01' and encapsulated in a BER-TLV data object with tag '87' and a length field coded according to [ISO 8825-1].
- The status word SW1 - SW2 of the unprotected APDU shall be contained in a BER-TLV data object with tag '99'.

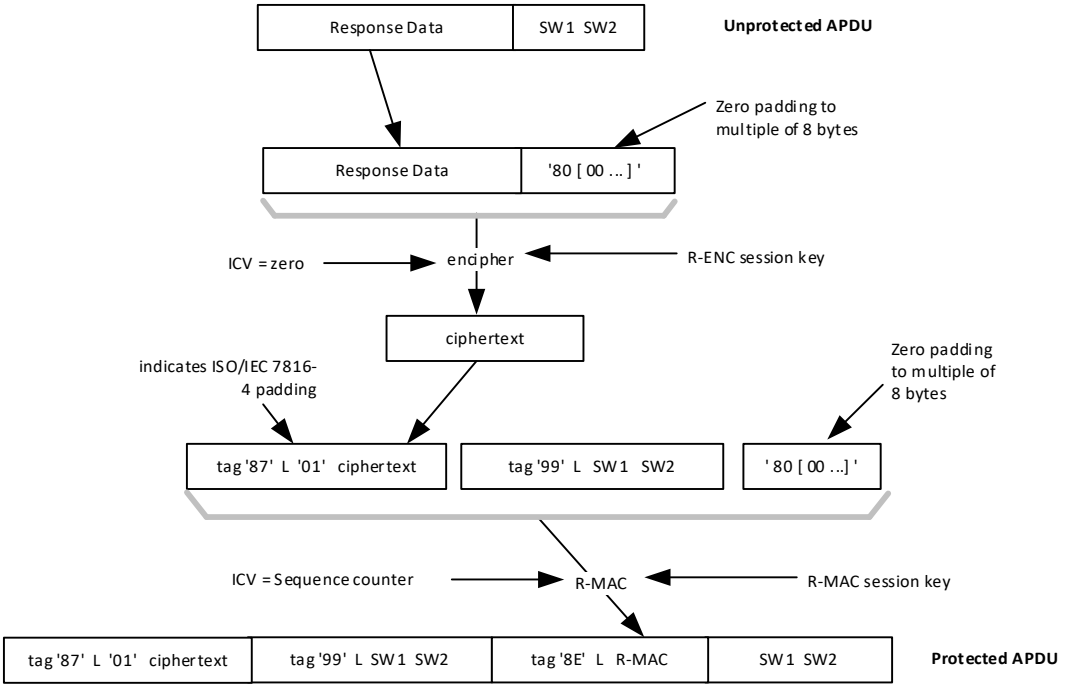
- The R-MAC shall be encapsulated in a BER-TLV data object with tag '8E' and appended at the end of the response data field.

No R-MAC padding is present in the transmitted APDU.

The Off-Card Entity, in order to verify the R-MAC, shall perform the same processing in order to generate an R-MAC and compare it with the transmitted R-MAC.

The following diagram shows the message reformatting that is performed by the card when a response message is protected for integrity and confidentiality.

Figure 4-6: Secure Messaging: Response Message Protected for Integrity and Confidentiality



4.2.7 Sensitive Data Encryption and Decryption

Data encryption is used when transmitting sensitive data to and from the card. For instance all keys transmitted to a card (e.g. in a PUT KEY command) should be encrypted. Data encryption is over and beyond the Current Security Level required for the Secure Channel Session. The encryption process uses the relevant data encryption session key (DEK) for sensitive data in command messages or for sensitive data in response messages. The encryption method uses AES in CBC mode depending on the Key Type of the DEK Key in the CRT; see [GPCS] section B.2.2. If the key type is omitted for the DEK Key it shall be known implicitly. The sensitive data block length shall be constructed as a multiple of 8-byte long block before the encryption operations: the eventual padding method is application specific.

The encryption is performed across the sensitive data and the result of each encryption becomes part of the encrypted data. This encrypted data becomes part of the clear text data field in the command/response message. The decryption is the exact opposite of the above operation: in particular, no padding is removed by the decryption operation.

5 Commands

Because certificates, digital signatures and some data fields can be long, command and response chaining as defined in [ISO 7816-4] is used to transfer successive data blocks.

With command chaining, the command data is sent in multiple APDUs, the command data being segmented arbitrarily. All except the final command in the chain shall indicate command chaining by setting to '1' bit 5 of the class byte according to [ISO 7816-4].

With response chaining, the response data is sent in multiple APDUs, the response data being segmented arbitrarily.

Table 5-1: SCP10 Command Support

Command	Secure Channel Initiation	
	Signature Without Message Recovery	Signature With Message Recovery
EXTERNAL AUTHENTICATE	✓	✓
GET CHALLENGE	✓	✓
GET DATA [certificate]	✓	✓
INTERNAL AUTHENTICATE	✓	✓
MANAGE SECURITY ENVIRONMENT	✓	✓
PERFORM SECURITY OPERATION [decipher]	✓	
PERFORM SECURITY OPERATION [verify certificate]	✓	✓

The following table summarizes the minimum security requirements for the APDU commands.

Table 5-2: Minimum Security Requirements for SCP10 commands

Command	Minimum Security
EXTERNAL AUTHENTICATE	Validated PK.OCE.AUT and card challenge
GET CHALLENGE	None
GET DATA [certificate]	None
INTERNAL AUTHENTICATE	Current Security Level is at least AUTHENTICATED or ANY_AUTHENTICATED
MANAGE SECURITY ENVIRONMENT	None
PERFORM SECURITY OPERATION [decipher]	None
PERFORM SECURITY OPERATION [verify certificate]	None

The following table provides the list of SCP10 command support per card Life Cycle State.

Table 5-3: SCP10 Command Support per Card Life Cycle State

Command	OP_READY			INITIALIZED			SECURED			CARD_ LOCKED	TERMINATED
	AM SD	DM SD	SD	AM SD	DM SD	SD	AM SD	DM SD	SD	SD	SD
EXTERNAL AUTHENTICATE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
GET CHALLENGE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
GET DATA [certificate]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
INTERNAL AUTHENTICATE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
MANAGE SECURITY ENVIRONMENT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
PERFORM SECURITY OPERATION [decipher]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
PERFORM SECURITY OPERATION [verify certificate]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	

Legend of Table 5-1 and Table 5-3

AM SD	Security Domain with Authorized Management privilege
DM SD	Supplementary Security Domain with Delegated Management privilege
SD	Other Security Domain
✓	Support required
Blank cell	Support optional
Striped cell	Support prohibited

5.1 EXTERNAL AUTHENTICATE Command

5.1.1 Definition and Scope

This command is used to authenticate the Off-Card Entity by the Security Domain. This command is also used with the key agreement option to support session key establishment. It shall be immediately preceded (on the same logical channel of the same card I/O interface) by a GET CHALLENGE command. It may be followed (on the same logical channel of the same card I/O interface) by an INTERNAL AUTHENTICATE command.

5.1.2 Command Message

The EXTERNAL AUTHENTICATE command message is coded as follows:

Table 5-4: EXTERNAL AUTHENTICATE Command Message

Code	Value	Meaning
CLA	'00' - '03', '40' - '4F', '10' - '13', or '50' - '5F'	See [GPCS] section 11.1.4
INS	'82'	EXTERNAL AUTHENTICATE
P1	'00'	Reference control parameter P1: no information given
P2	'00'	Reference control parameter P2: no information given
Lc	'xx'	Length of Entity Signature (key transport) or encrypted Off-Card Entity Signature (key agreement)
Data	'xx xx…'	Command data field
Le	-	Not present

A Security Domain may support other values of Reference Control Parameters P1 and P2 as defined in [ISO 7816-4].

5.1.3 Data Field Sent in the Command Message

The data field of the EXTERNAL AUTHENTICATE command message contains the Off-Card Entity Signature (key transport) or encrypted Off-Card Entity Signature (key agreement) – see section 2.4.4, Off-Card Entity Authentication, for further details.

5.1.4 Data Field Returned in the Response Message

No data is returned from this command.

5.1.5 Processing State Returned in the Response Message

A successful execution of the command shall be indicated by status bytes '90' '00'.

This command may return either a general error condition as listed in [GPCS] section 11.1.3, General Error Conditions, or one of the following specific errors and warning conditions.

Table 5-5: Error Conditions

SW1	SW2	Meaning
'63'	'00'	Verification of certificate failed
'94'	'84'	Algorithm not supported

5.2 GET CHALLENGE Command

5.2.1 Definition and Scope

This command is used to obtain a random challenge from the Security Domain, to support authentication of the Off-Card Entity to the Security Domain. It precedes the EXTERNAL AUTHENTICATE command. It shall have been preceded (on the same logical channel of the same card I/O interface), immediately or not, by a MANAGE SECURITY ENVIRONMENT or PERFORM SECURITY OPERATION [verify certificate] command.

5.2.2 Command Message

The GET CHALLENGE command message is coded according to the following table:

Table 5-6: GET CHALLENGE Command Message

Code	Value	Meaning
CLA	'00' - '03' or '40' - '4F'	See [GPCS] section 11.1.4
INS	'84'	GET CHALLENGE
P1	'00'	Reference Control Parameter P1: no information given
P2	'00'	Reference Control Parameter P2: no information given
Lc	Absent	
Data	Absent	
Le	'00'	

A Security Domain may support other values of Reference Control Parameters P1 and P2 as defined in [ISO 7816-4].

5.2.3 Data Field Sent in the Command Message

There is no command data.

5.2.4 Data Field Returned in the Response Message

Data returned comprises a card challenge, coded on 8 bytes with the key agreement option and 16 bytes with the key transport option.

5.2.5 Processing State Returned in the Response Message

A successful execution of the command shall be indicated by status bytes '90' '00'.

The command may return a general error condition as listed in [GPCS] section 11.1.3, General Error Conditions.

5.3 GET DATA [certificate] Command

5.3.1 Definition and Scope

The GET DATA [certificate] command is used to retrieve either information about all certificates that can be retrieved from the card, or a single certificate. This command may be issued at any time, in particular it may be interleaved with PERFORM SECURITY OPERATION [verify certificate] commands. If it is issued when a Secure Channel Session is active, it must comply with the Current Security Level of that Secure Channel Session.

5.3.2 Command Message

The following command is used to obtain certificate information or a single certificate from the Security Domain.

Table 5-7: GET DATA [certificate] Command Message

Code	Value	Meaning
CLA	'00' - '0F', '40' - '4F', '60' - '6F', '80' - '8F', 'C0' - 'CF', or 'E0' - 'EF'	See [GPCS] section 11.1.4
INS	'CA' or 'CB'	If CLA = '00' - '0F', '40' - '4F', or '60' - '6F', even or odd instruction code 'CA' or 'CB' If CLA = '80' - '8F', 'C0' - 'CF', or 'E0' - 'EF', even instruction code 'CA'
P1 P2		Reference Control Parameters P1 and P2: there are three options:
	'7F 21'	Tag of certificate
	'50 31'	Data object identifier of EF.OD
	'xx xx'	(Any other value) data object identifier of a certificate
Lc	'xx' or omitted	Not present if P1 P2 = '7F 21', otherwise length of command data
Data	'xx xx...' or omitted	Not present, or command data
Le	'00'	

5.3.3 Reference Control Parameters P1 and P2

If P1 P2 = '7F 21', the command is a request to retrieve the certificate of the Security Domain's default public key, CERT.SD.AUT.

Otherwise, for any value other than those defined in [GPCS] section 11.3, GET DATA Command, the command is a request to access EF.OD or another certificate and the instruction code shall be set to 'CA' if the class byte indicates a GlobalPlatform command (CLA set to '80' - '8F', 'C0' - 'CF', or 'E0' - 'EF') or 'CB' if the class byte indicates an ISO command (CLA set to '00' - '0F', '40' - '4F', or '60' - '6F').

- If P1 P2 = '50 31', the command is a request to retrieve details of all available certificates held by the Security Domain for retrieval and verification by an Off-Card Entity.
- Otherwise, for any other value of P1 P2, the command shall be treated as a request to retrieve a certificate whose pointer is given in P1 P2. This would typically follow a command with P1 P2 = '50 31', where the Cryptographic Information Objects returned in the response have pointers to individual certificates. However, the Off-Card Entity may already know the location of a required certificate, and issue this command directly.

5.3.4 Data Field Sent in the Command Message

When P1-P2 is different from '7F21', the command data shall be present and coded as follows:

Table 5-8: GET DATA [certificate] Command Data Message

Name	Length	Name	Presence
Tag list tag	1	'5C'	Mandatory
Tag list length	1	'00' (empty, indicating 'retrieve all data')	Mandatory

5.3.5 Data Field Returned in the Response Message

When the command is issued to retrieve a certificate (P1-P2 different from '5031') and the class byte indicates a GlobalPlatform command (CLA set to '80' - '8F', 'C0' - 'CF', or 'E0' - 'EF'), the certificate shall be returned TLV-coded as follows:

Table 5-9: GET DATA [certificate] Response Data Field – Certificate

Name	Length	Name	Presence
Certificate tag	2	'7F21'	Conditional
Certificate length	1, 2, or 3	'00' - '7F', or '81 80' - '81 FF', or '82 01 00' - '82 FF FF'	Conditional
Certificate data	n	'xxxx...' (as defined in section 2.3.4).	Mandatory

When the command is issued to retrieve a certificate (P1-P2 different from '5031') and the class byte indicates an ISO command (CLA set to '00' - '0F', '40' - '4F', or '60' - '6F'), only the certificate value shall be returned.

If the command was issued to retrieve certificate information details (P1-P2 = '5031'), the contents of EF.OD listing all the Cryptographic Information Objects for the different certificates held in by the Security Domain shall be returned as defined in section 2.3.3.

5.3.6 Processing State Returned in the Response Message

A successful execution of the command shall be indicated by status bytes '90' '00'.

This command may return either a general error condition as listed in [GPCS] section 11.1.3, General Error Conditions, or one of the following errors and warning conditions:

Table 5-10: Error Conditions

SW1	SW2	Meaning
'6A'	'80'	Incorrect values in command data
'6A'	'88'	Referenced data not found

5.4 INTERNAL AUTHENTICATE Command

5.4.1 Definition and Scope

This command is used to authenticate the Security Domain, by the Off-Card Entity. This command is also used with the key agreement option, to support session key establishment. It shall be immediately preceded (on the same logical channel of the same card I/O interface) by an EXTERNAL AUTHENTICATE command.

5.4.2 Command Message

The INTERNAL AUTHENTICATE command message is coded according to the following table:

Table 5-11: INTERNAL AUTHENTICATE Command Message

Code	Value	Meaning
CLA	'00' - '03' or '40' - '4F'	See [GPCS] section 11.1.4
INS	'88'	INTERNAL AUTHENTICATE
P1	'00'	Reference control parameter P1: no information given
P2	'00'	Reference control parameter P2: no information given
Lc	'xx'	Length of Off-Card Entity challenge
Data	'xx xx…'	Command data field
Le	'00'	

A Security Domain may support other values of Reference Control Parameters P1 and P2 as defined in [ISO 7816-4].

5.4.3 Data Field Sent in the Command Message

The data field of the INTERNAL AUTHENTICATE command message contains the following data:

Table 5-12: INTERNAL AUTHENTICATE Command Data Field

Name	Length	Name	Presence
Off-Card Entity challenge	8 or 16	'xxxx...'	Mandatory
Off-Card Entity id	8	'xxxx...' (part of CERT.OCE.AUT)	Mandatory

Off-Card Entity challenge is 16 bytes for the key transport option, 8 bytes for the key agreement option.

5.4.4 Data Field Returned in the Response Message

The Card Signature (key transport) or encrypted Card Signature (key agreement) is returned – see section 2.4.3 for details.

5.4.5 Processing State Returned in the Response Message

A successful execution of the command shall be indicated by status bytes '90' '00'.

This command may return either a general error condition as listed in [GPCS] section 11.1.3, General Error Conditions, or one of the following specific errors and warning conditions:

Table 5-13: Warning Conditions

SW1	SW2	Meaning
'61'	'xx'	Response data incomplete, 'xx' more bytes available

Table 5-14: Error Conditions

SW1	SW2	Meaning
'6A'	'80'	Incorrect values in command data

5.5 MANAGE SECURITY ENVIRONMENT Command

5.5.1 Definition and Scope

This command selects the Secure Channel Protocol '10' and its options as well as defining specific keys to be used by the Security Domain. If a Secure Channel Session is active on this logical channel and card I/O interface, it shall be terminated on receipt of this command, regardless of the validity of the command.

5.5.2 Command Message

The MANAGE SECURITY ENVIRONMENT command message is coded according to the following table:

Table 5-15: MANAGE SECURITY ENVIRONMENT Command Message

Code	Value	Meaning
CLA	'00' - '03' or '40' - '4F'	See [GPCS] section 11.1.4
INS	'22'	MANAGE SECURITY ENVIRONMENT
P1	'81' or 'C1'	Reference Control Parameter P1 '81': External (Off-Card Entity) Authentication only 'C1': External and Internal (Mutual) Authentication
P2	'A4' or 'B6'	Reference Control Parameter P2: 'A4': Authentication: no certificate verification will be performed by the card 'B6': Digital signature: certificate verification will be performed by the card
Lc	'xx'	Length of command data
Data	'xx xx…'	Off-Card Entity data
Le	-	Not present

5.5.3 Reference Control Parameter P1

The value of P1 is based on [ISO 7816-4], as follows:

Table 5-16: MANAGE SECURITY ENVIRONMENT Reference Control Parameter P1

b8	b7	b6	b5	b4	b3	b2	b1	Description
1	-	-	-	-	-	-	-	Verification, encipherment, external authentication and key agreement
-	1	-	-	-	-	-	-	Computation, decipherment, internal authentication and key agreement
-	-	-	-	-	-	-	1	SET
-	-	X	X	X	X	X	-	Values defined in [ISO 7816-4]

5.5.4 Reference Control Parameter P2

This is set according to the template that is appropriate for the subsequent message flow, as defined in [ISO 7816-4]: 'A4' if certificate verification by the card is omitted, 'B6' if certificate verification is to be performed by the card.

5.5.5 Data Field Sent in the Command Message

The command data field is formatted as follows:

Table 5-17: MANAGE SECURITY ENVIRONMENT Command Data Field

Tag	Length	Name	Presence
'80'	2	Cryptographic mechanism reference: SCP id + options “i” (= scp || i)	Mandatory
'83'	0 or 1-n	Public key reference	Conditional
'84'	0 or 1-n	Private key reference	Conditional

The ‘cryptographic mechanism reference’ (tag '80', value '10') designates GlobalPlatform asymmetric Secure Channel Protocol '10' (SCP10) and its options, and distinguishes it from any other protocol that might be supported by the selected Issuer Security Domain, Security Domain or Application. Option “i” is described in section 2.1.

The ‘public key reference’ (tag '83') designates the public key to be used by the Security Domain in subsequent cryptographic operations during Secure Channel Session initiation. The referenced public key shall have already been validated by the Security Domain or shall be the public key of the Security Domain’s Trust Point for External Authentication. If no reference value is provided in the command, the Security Domain shall use by default the public key of its Trust Point for External Authentication, designated PK.TP_EX.AUT, as the first key to use for certificate verification within a session; see section 2.3.1.

The ‘private key reference’ (tag '84') designates the private key to be used by the Security Domain in subsequent cryptographic operations during Secure Channel Session initiation. The referenced private key shall be present and known to the Security Domain. If no reference value is provided in the command, the Security Domain shall use by default its private key designated SK.SD.AUT

A Security Domain may support other data elements as defined in [ISO 7816-4].

5.5.6 Data Field Returned in the Response Message

No data is returned from this command.

5.5.7 Processing State Returned in the Response Message

A successful execution of the command shall be indicated by status bytes '90' '00'.

This command may return either a general error condition as listed in [GPCS] section 11.1.3, General Error Conditions, or one of the following specific errors and warning conditions.

Table 5-18: Error Conditions

SW1	SW2	Meaning
'6A'	'88'	Referenced data not found
'94'	'84'	Algorithm not supported

5.6 PERFORM SECURITY OPERATION [decipher] Command

5.6.1 Definition and Scope

This command is used with the session key transport option, and transmits the AES session keys from the Off-Card Entity to the Security Domain. It shall have been preceded (on the same logical channel of the same card I/O interface), immediately or not, by a MANAGE SECURITY ENVIRONMENT or PERFORM SECURITY OPERATION [verify certificate] command.

5.6.2 Command Message

The PERFORM SECURITY OPERATION [decipher] command message is coded according to the following table:

Table 5-19: PERFORM SECURITY OPERATION [decipher] Command Message

Code	Value	Meaning
CLA	'00' - '03', '40' - '4F', '10' - '13', or '50' - '5F'	See [GPCS] section 11.1.4
INS	'2A'	PERFORM SECURITY OPERATION [decipher]
P1	'80'	Reference Control Parameter P1: clear text object
P2	'84'	Reference Control Parameter P2: cryptogram (plain value encoded in BER-TLV) present in the command
Lc	'xx'	Length of encrypted data
Data	'xx xx…'	Encrypted data
Le	-	Not present

A Security Domain may support other values of Reference Control Parameters P1 and P2 as defined in [ISO 7816-4].

5.6.3 Data Field Sent in the Command Message

The data field of the PERFORM SECURITY OPERATION [decipher] command message contains the Encrypted Off-Card Entity Session Data.

The Off-Card Entity session keys are in control reference templates as defined in section 4.1.2, one template per session key, concatenated for encryption. Off-Card Entity Session Key Data is formed by padding the session key CRTs to the length of the Security Domain public key modulus as shown in the table below, and encrypting the result with the Security Domain public key (PK.SD.AUT).

Table 5-20: Off-Card Entity Session Key Data – Clear Text before Encryption

Meaning	Length	Meaning	Presence
Padding	2	'0002'	
Padding	8-n	'FF'...'FF'	
Padding	1	'00'	
Tag of Security Level ('D3')	1	'D3'	Mandatory
Length of Security Level	1	'01'	Mandatory
Security Level	1	'xx'	Mandatory
CRT tag	1	'B4' (CCT) or 'B8' (CT)	Mandatory
Length of CRT	1	'00' - '7F'	Mandatory
CRT contents (with session key)	n	'xxxx...'	Mandatory
…..	….	…..	…
CRT tag	1	'B4' (CCT) or 'B8' (CT)	Optional
Length of CRT	1	'00' - '7F'	Conditional
CRT contents (with session key)	n	'xxxx...'	Conditional

5.6.4 Data Field Returned in the Response Message

No data is returned from this command.

5.6.5 Processing State Returned in the Response Message

A successful execution of the command shall be indicated by status bytes '90' '00'.

This command may return either a general error condition as listed in [GPCS] section 11.1.3, General Error Conditions, or one of the following specific errors and warning conditions.

Table 5-21: Error Conditions

SW1	SW2	Meaning
'6A'	'80'	Incorrect values in command data

5.7 PERFORM SECURITY OPERATION [verify certificate] Command

5.7.1 Definition and Scope

This command is used to provide a certificate to the Security Domain for verification. It may be preceded (on the same logical channel of the same card I/O interface) by a MANAGE SECURITY ENVIRONMENT command or a PERFORM SECURITY OPERATION [verify certificate] command and may be interleaved with GET DATA [certificate] commands.

5.7.2 Command Message

The PERFORM SECURITY OPERATION [verify certificate] command message is coded according to the following table:

Table 5-22: PERFORM SECURITY OPERATION [verify certificate] Command Message

Code	Value	Meaning
CLA	'00' - '03', '40' - '4F', '10' - '13' or '50' - '5F'	See [GPCS] section 11.1.4
INS	'2A'	PERFORM SECURITY OPERATION [verify certificate]
P1	'00'	Reference Control Parameter P1: no object in the response
P2	'AE' or 'BE'	Reference Control Parameter P2: input template for certificate verification present in the command 'AE': non-self descriptive card verifiable certificate (only the concatenated value fields are certified) 'BE': self-descriptive card verifiable certificate (the TLV data elements are certified)
Lc	'xx'	Length of certificate data
Data	'xx xx…'	Certificate data
Le	-	Not present

5.7.3 Data Field Sent in the Command Message

The command data field is certificate data as follows.

Table 5-23: PERFORM SECURITY OPERATION [verify certificate] Command Data Field

Name	Length	Name	Presence
Certificate tag	2	'7F21'	Mandatory
Certificate length	1, 2, or 3	'00' - '7F', or '81 80' - '81 FF' or '82 01 00' - '82 FF FF'	Mandatory
Certificate data	n	'xxxx...' (as described in section 2.3.4)	Mandatory

The Security Domain verifies the certificate presented using the Current Public Key as known to the Security Domain. For the first certificate presented in the session, the Current Public Key is either the default Public Key - the Public Key of the Trust Point, or another Public Key as announced in the MANAGE SECURITY ENVIRONMENT command. A series of certificates can be presented to the Security Domain in subsequent PERFORM SECURITY OPERATION [verify certificate] commands. For each subsequent PERFORM SECURITY OPERATION [verify certificate] command, the Current Public Key is the one that was certified in the certificate presented in the previous PERFORM SECURITY OPERATION [verify certificate] command.

5.7.4 Data Field Returned in the Response Message

No data is returned from this command.

5.7.5 Processing State Returned in the Response Message

A successful execution of the command shall be indicated by status bytes '90' '00'.

This command may return either a general error condition as listed in [GPCS] section 11.1.3, General Error Conditions, or one of the following specific errors and warning conditions.

Table 5-24: Error Conditions

SW1	SW2	Meaning
'63'	'00'	Verification of the certificate failed
'68'	'83'	The last command of the chain was expected
'6A'	'80'	Incorrect values in command data