GLOBALPLATFORM®

THE STANDARD FOR MANAGING APPLICATIONS ON SECURE CHIP TECHNOLOGY

# GlobalPlatform Card

# Secure Channel Protocol '11'

## Card Specification v2.2 – Amendment F

Version 1.0

Public Release

May 2015

Document Reference: GPC_SPE_093

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 new secure channel protocol, named **Secure Channel Protocol '11' (SCP11)**, based on Elliptic Curve Cryptography (ECC) for mutual authentication and secure channel initiation and on AES for secure messaging.

## 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://www.globalplatform.org/specificationsipdisclaimers.asp. 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

**Table 1-1: Normative References**

| Standard / Specification | Description | Ref |
|---|---|---|
| GlobalPlatform Card Specification | GlobalPlatform Card Specification v2.2.1, January 2011 | [GPCS] |
| GPCS Amendment D | GlobalPlatform Card Technology, Secure Channel Protocol 03, Card Specification v 2.2 – Amendment D, v1.1.1 | [Amd D] |
| GPCS Amendment E | GlobalPlatform Card Technology, Security Upgrade for Card Content Management, Card Specification v2.2 – Amendment E, v1.0.1 | [Amd E] |
| BSI TR-03111, Version 2.0 | BSI Technical Guideline TR-03111: Elliptic Curve Cryptography | [TR 03111] |
| NIST SP 800-56A Revision 2 | Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography, May 2013 | [NIST 800-56A] |

**Table 1-2: Informative References**

| Standard / Specification | Description | Ref |
|---|---|---|
| GPCS Amendment A | GlobalPlatform Card Technology, Confidential Card Content Management, Card Specification v2.2 – Amendment A, v1.0.1 | [Amd A] |
| Trusted User Interface API | GlobalPlatform Device Technology, Trusted User Interface API, v1.0 | [TUI] |

## 1.4 Terminology and Definitions

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

## 1.5 Abbreviations and Notations

Abbreviations and notations used in this document are included in Table 1-3.

**Table 1-3: Abbreviations and Notations**

| Abbreviation / Notation | Meaning |
|---|---|
| AES | Advanced Encryption Standard |
| APDU | Application Protocol Data Unit |
| API | Application Programming Interface |
| BCD | Binary Coded Decimal |
| BER | Basic Encoding Rules |
| CA | Controlling Authority |
| CA-KLCC | Controlling Authority for Confidential Key Loading Card Certificates |
| CA-KLOC | Controlling Authority for Confidential Key Loading OCE Certificates |
| C-DECRYPTION | Command Decryption |
| CERT.OCE.ECKA | Certificate containing the public key of the OCE used for key agreement |
| CERT.SD.ECKA | Certificate containing the public key of the SD used for key agreement |
| CLA | CLAss byte of command message |
| C-MAC | Command MAC |
| CRT | Control Reference Template |
| DGI | Data Grouping Identifier |
| EC | Elliptic Curve |
| ECC | Elliptic Curve Cryptography |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| ECKA | Elliptic Curve Key Agreement |
| ePK.OCE.ECKA | Ephemeral public key of the OCE used for key agreement |
| ePK.SD.ECKA | Ephemeral public key of the SD used for key agreement |

| Abbreviation / Notation | Meaning |
|---|---|
| eSK.OCE.ECKA | Ephemeral private key of the OCE used for key agreement |
| eSK.SD.ECKA | Ephemeral private key of the SD used for key agreement |
| INS | INStruction byte of command message |
| Key-DEK | Data Encryption Key |
| KID | Key Identifier |
| KVN | Key Version Number |
| 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 |
| MOC | Mandatory, Optional, Conditional |
| OCE | Off Card Entity |
| P1 | Reference control Parameter 1 |
| P2 | Reference control Parameter 2 |
| PIN | Personal Identification Number |
| PK.CA-KLCC.ECDSA | Public key of the CA-KLCC used for verifying certificates |
| PK.CA-KLOC.ECDSA | Public key of the CA-KLOC used for verifying certificates |
| PK.OCE.ECKA | Public key of the OCE used for key agreement |
| PK.SD.ECKA | Public key of the SD used for key agreement |
| R-ENCRYPTION | Response Encryption |
| R-MAC | Response MAC |
| RFU | Reserved for Future Use |
| SCP | Secure Channel Protocol |
| SD | Security Domain |
| S-DEK | Session Data Encryption Key |
| S-ENC | Secure Channel session key for command and response encryption |
| ShS | Shared Secret (concatenated ShSe and ShSs) |
| ShSe | Shared Secret calculated from the ephemeral keys |
| ShSs | Shared Secret calculated from at least one static keys |
| SK.CA-KLOC.ECDSA | Private key of the OCE used for signing certificates |
| SK.OCE.ECKA | Private key of the OCE used for key agreement |
| SK.SD.ECKA | Private key of the SD used for key agreement |
| S-MAC | Secure Channel C-MAC session key |
| S-RMAC | Secure Channel R-MAC session key |

| Abbreviation / Notation | Meaning |
|---|---|
| TLV | Tag Length Value |
| UTF-8 | Unicode Transformation Format – 8-bit |
| Var | Variable |

## 1.6 Revision History

**Table 1-4: Revision History**

| Date | Version | Description |
|---|---|---|
| May 2015 | 1.0 | Initial Release |

# 2 Secure Channel Protocol '11'

## 2.1 Scope of the Document

This document specifies a new secure channel protocol, named **Secure Channel Protocol '11' (SCP11)**, based on Elliptic Curve Cryptography (ECC) for mutual authentication and secure channel initiation and on AES for secure messaging.

It reuses cryptographic mechanisms defined in GPCS Amendment E: Security Upgrade for Card Content Management [Amd E] and in GPCS Amendment D: SCP03 [Amd D].

Two variants of the protocol are defined:

- SCP11a, providing mutual authentication between the Off Card Entity (OCE) and the card.
- SCP11b, providing authentication of the card to the OCE only. Authentication of the OCE to the card has to be provided by other means; an example is provided in Annex A.

## 2.2 Use Cases and Requirements

Compared to SCP03, this protocol allows authentication and secure channel initiation based on certificates instead of pre-shared keys. This provides greater flexibility in cases where the two entities setting up the secure channel are not deployed in strict pairs.

ECC provides suitable security strength for the establishment of session keys for all three variants of AES: AES-128, AES-192, and AES-256.

# 3 Specification Amendments

## 3.1 Algorithms

This specification combines algorithms already specified in [Amd D] and [Amd E]. However, SCP11 uses different input data for ECKA and the Key Derivation compared to [Amd E].

### 3.1.1 ECKA

An Elliptic Curve Key Agreement Algorithm (ECKA) is used in this specification for the establishment of session keys. A description of such schemes can be found e.g. in [TR 03111].

ECKA used in this specification shall follow the definition for the Key Agreement Algorithm in [TR 03111]. The algorithm is executed twice, once with ephemeral keys and once with static keys (SCP11a) or with an ephemeral and a static key (SCP11b). The scheme for SCP11a is equivalent to the scheme named "(Cofactor) Full Unified Model, C(2, 2, ECC CDH)" in [NIST 800-56A] for curves with a cofactor of 1. The recommendations in [NIST 800-56A] on the handling of ephemeral keys and of intermediate results (e.g. the shared secrets ShSe and ShSs) should be taken into account in an implementation.

> ***Note:*** *Performing all the checks specified in [TR 03111] (including the check that the secret points are not zero) is required to avoid attacks on ephemeral public keys.*

### 3.1.2 Key Derivation

The shared secret ShS generated by Key Agreement Algorithm is not used directly as a key for cryptographic operations, but as an input to a key derivation process.

A key for calculating a receipt and the session keys are derived from the shared secret as defined in [TR 03111] for the "X9.63 Key Derivation Function". This key derivation includes additional information, the "SharedInfo" of the key derivation algorithm.

## 3.2 Controlling Authority Roles

Within the context of SCP11, the Controlling Authority (CA) has two different roles:

- Providing certificates for the SD: CERT.SD.ECKA
- Providing certificates for the OCE: CERT.OCE.ECKA

As there is no technical need that one actor provides both roles, those roles are distinguished in this document:

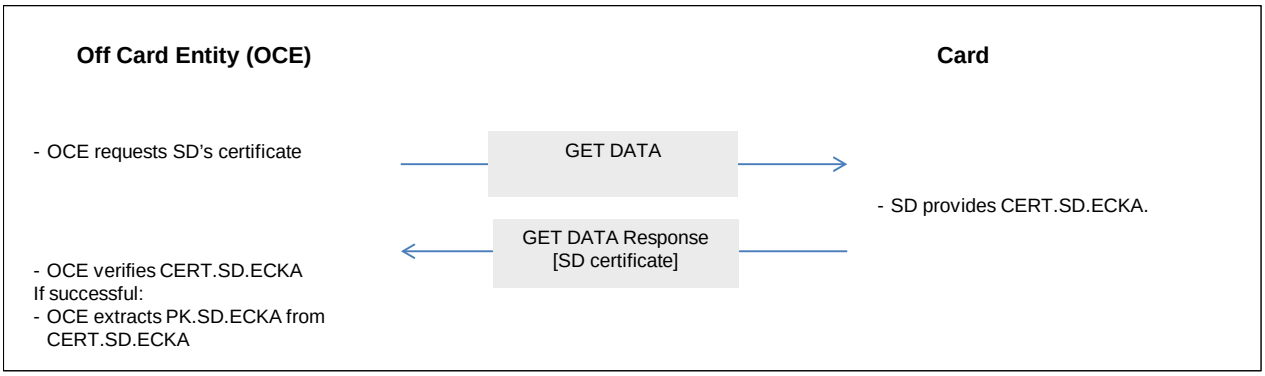
- CA-KLCC denotes the role providing certificates for the SD on the Card: Controlling Authority for Confidential Key Loading Card Certificates.
- CA-KLOC denotes the role providing certificates for the OCE: Controlling Authority for Confidential Key Loading OCE Certificates.

# 4 Secure Channel Protocol Usage

## 4.1 Protocol Overview

Before setting up a secure channel, the OCE has to be in the possession of the SD's certificate. The certificate may be retrieved from the SD as shown below. The OCE may store the certificate (or parts of it) for use in future secure channel sessions. The OCE may also be provided with the certificate by some other means.

**Figure 4-1: Initial Certificate Retrieval**




The following two figures provide an overview of the two variants of SCP11.

SCP11a provides mutual authentication of the OCE and the SD. For this purpose, the OCE has to provide the SD with its certificate in a PERFORM SECURITY OPERATION command prior to the establishment of the secure channel. Dependent on the implementation option, the SD may store the public key extracted from the certificate persistently.

- If the SD stores the public key persistently, it is not required that a PERFORM SECURITY OPERATION command is sent immediately before the MUTUAL AUTHENTICATE command – other commands may be interleaved. The certificate can be used in multiple future secure channel sessions, even in new card sessions after a power down.
- If the SD does not store the public key persistently, the MUTUAL AUTHENTICATE command has to immediately follow the PERFORM SECURITY OPERATION command.

As SCP11b provides only authentication of the SD to the OCE, the PERFORM SECURITY OPERATION is not required.

**Figure 4-2: SCP11a Protocol Overview**

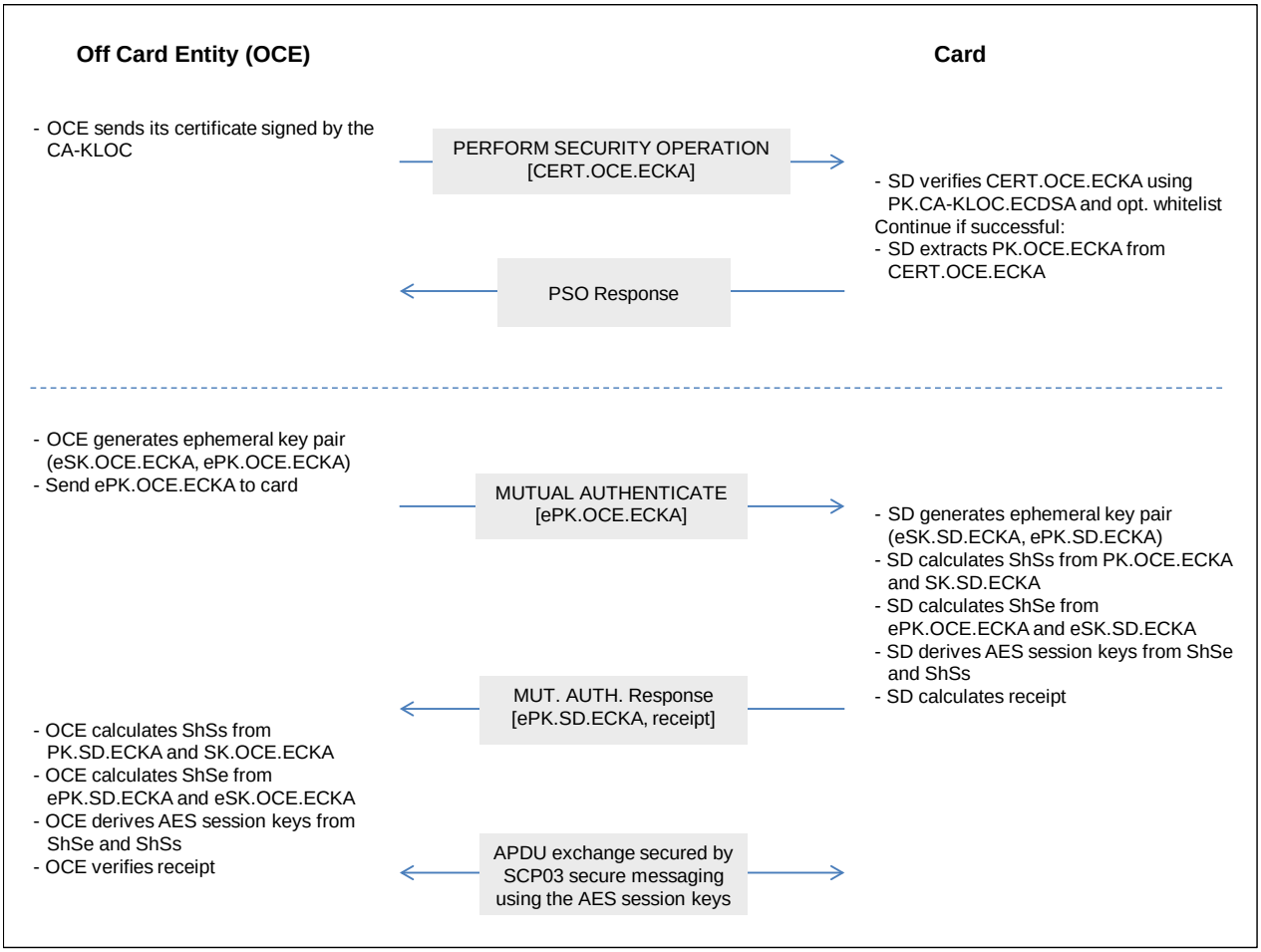
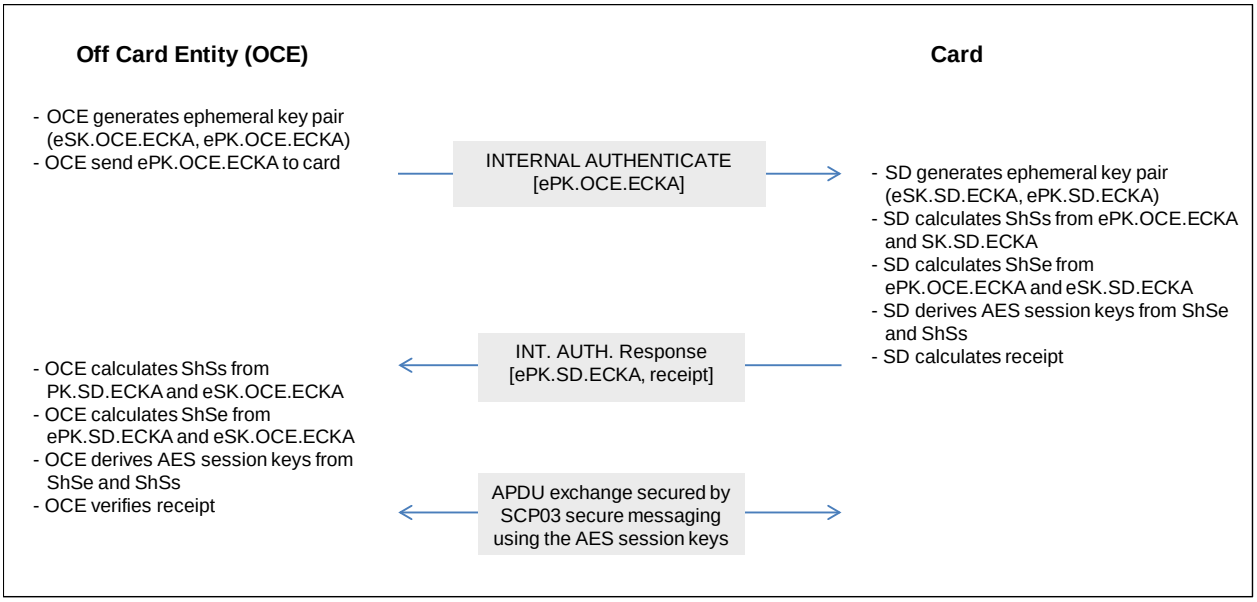



**Figure 4-3: SCP11b Protocol Overview**



## 4.2 Secure Communication Configuration

Three levels of security are supported by SCP11:

- Authentication: Assurance that the peer entity is in fact the entity it claims to be
- Integrity and data origin authentication
- Confidentiality

Details for SCP11a and SCP11b are given in section 4.3 and section 4.4.

In SCP11 the implementation option “i” is formed as a bitmap on one byte as follows:

**Table 4-1: Values of Parameter “i”**

| b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | Description |
|---|---|---|---|---|---|---|---|---|
| | X | X | X | X | | | | RFU (set to 0) |
| | | | | | | | X | 1: SCP11a supported |
| | | | | | | X | | 1: SCP11b supported |
| | | | | | X | | 1 | 1: SD persistently stores PK.OCE.ECKA (only applicable to SCP11a) |
| X | | | | | | | | Reserved; see section E.1.1 of [GPCS] |

An implementation may support one or both variants of SCP11.

Applications can retrieve the current security level via the API method *getSecurityLevel()* to find out which variant is used during a session. A current security level of AUTHENTICATED or ANY_AUTHENTICATED indicates that mutual authentication was successful and that SCP11a is used currently (see section 4.6).

If the Security Level does not indicate AUTHENTICATED, then an SD shall only accept SELECT, GET DATA, and the commands for initiating a secure channel.

## 4.3 Authentication

Authentication is achieved through the process of initiating a Secure Channel and provides assurance to an entity that it is communicating with an authenticated entity.

For SCP11a only: The OCE authenticates to the SD by providing a certificate signed by the CA-KLOC and by providing the first APDU after secure channel establishment with a correct MAC. If a whitelist with one or more Certificate Serial Number entries exists in the SD for the CA-KLOC's public key, the SD also verifies that the certificate is contained in a whitelist. Else the SD accepts all certificates signed by the CA-KLOC.

Use of the whitelist can provide the following benefits:

- A strong binding to one (or multiple) OCE(s)
- Protection against compromised OCEs

It is recommended to use the whitelist also as a revocation mechanism for OCE certificates.

The SD authenticates to the OCE by providing a certificate signed by the CA-KLCC and by generating a receipt at the end of the key establishment procedure. Implementation of a revocation mechanism for the SD's certificate is recommended (e.g. by the OCE using a white- or a blacklist for the SD certificates), but out of scope of this specification. Such a mechanism can protect the system against compromised SD keys.

SCP11a provides Mutual Authentication between the OCE and the SD.

SCP11b provides Authentication of the SD to the OCE only.

## 4.4 Message Integrity and Data Confidentiality

Message Integrity and Data Confidentiality is achieved by the secure messaging as defined in [Amd D], which is applied to all APDUs following the MUTUAL or INTERNAL AUTHENTICATE command.

Changing of the security level with the commands BEGIN or END R-MAC SESSION defined in [Amd D] shall not be supported by SCP11.

## 4.5 Forward Secrecy

Both variants of SCP11 provide Forward Secrecy (sometimes also called Perfect Forward Secrecy).

This is achieved by the ephemeral key pairs generated by the OCE and the SD which are used only once for the establishment of the session keys and which are destroyed immediately thereafter.

Forward secrecy assures the continued confidentiality of the data exchanged in a session even if the static private keys are compromised at a later point in time.

## 4.6 API and Security Level

An SD supporting SCP11 shall implement the `SecureChannel` interface of the API specified in [GPCS]. An application associated to the SD may use this API to request the SD to handle the SCP11 specific protocol.

The following APDUs are handled by the *processSecurity()* method of the SD: PERFORM SECURITY OPERATION, MUTUAL AUTHENTICATE, and INTERNAL AUTHENTICATE.

A call to *decryptData()* or *encryptData()* shall throw an ISOException with reason code '6985' if Key-DEK is personalized, but not available for the calling application due to the settings in the Key Access Coding (see section 5.3.3). The secure channel session shall not be aborted.

> ***Note:*** *The `SecureChannelx` interface is not supported, as BEGIN or END R-MAC SESSION is not supported by SCP11. Support of the `SecureChannelx2` interface is out of scope of this specification.*

The following shall apply for the Security Level:

The Current Security Level of a communication not included in a Secure Channel Session shall be set to NO_SECURITY_LEVEL.

The Current Security Level established in a Secure Channel Session is a bitmap combination of the following values: AUTHENTICATED, ANY_AUTHENTICATED, C_MAC, R_MAC, C_DECRYPTION, and R_ENCRYPTION.

The Current Security Level shall be set as follows:

- NO_SECURITY_LEVEL when a Secure Channel Session is terminated or not yet fully initiated;
- For SCP11a, C_MAC, R_MAC, and either AUTHENTICATED or ANY_AUTHENTICATED after a successful processing of an MUTUAL AUTHENTICATE command;
- For SCP11b, C_MAC and R_MAC after a successful processing of an INTERNAL AUTHENTICATE command;
- C_DECRYPTION and R_ENCRYPTION in addition after a successful processing of an MUTUAL or INTERNAL AUTHENTICATE command with key usage qualifier set to '3C'.

> ***Note:*** *The key usage qualifier contained in the command data of the MUTUAL or INTERNAL AUTHENTICATE command is used to determine the security level of the secure channel session.*

As defined in [GPCS] section 10.4.2, ANY_AUTHENTICATED is the security level achieved if any OCE not being the owner of the SD authenticates using asymmetric cryptography. AUTHENTICATED is achieved if the owner of the SD or Application authenticates. The SD identifies the owner by the Subject Identifier (TLV with tag '5F20') in the OCE certificate matching the Application Provider Identifier of the SD or Application, which was provided as a parameter (TLV with tag '5F20' within the CRT TLV with tag 'B6') in the INSTALL [for install] command and which cannot be changed subsequently.

> ***Note:*** *The CRT (tag 'B6') containing the Application Provider Identifier (tag '5F20') serves two purposes: To provide the Application Provider Identifier to the SD to be used in secure channel protocols with asymmetric cryptography and to provide token information for Delegated Management.*

## 4.7 Protocol Rules

In accordance with the general rules described in section 10 of [GPCS], the following protocol rules apply to SCP11:

- The successful initiation of a Secure Channel Session shall set the Current Security Level to the security level indicated in the MUTUAL or INTERNAL AUTHENTICATE command.
- The Current Security Level shall apply to the entire Secure Channel Session.
- When the Current Security Level is set to NO_SECURITY_LEVEL:
  - 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 to apply its own security rules.
- If a Secure Channel Session is active (i.e. Current Security Level different from NO_SECURITY_LEVEL), the security of the incoming command shall be checked according to the Current Security Level regardless of the command secure messaging indicator:
  - When the security of the command does not match 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;
  - In all other cases, the Secure Channel Session shall remain active and the Current Security Level unmodified. The Application is responsible for further processing the command.
- If a Secure Channel Session is aborted, it is still considered not terminated;
- The current Secure Channel Session shall be terminated (if aborted or still open) and the Current 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).

# 5 Cryptographic Keys

## 5.1 ECC Keys

**Table 5-1: ECC Keys**

| Key | Usage | Length | Remark |
|---|---|---|---|
| eSK.SD.ECKA | Ephemeral private key of the SD used for key agreement | see below | Mandatory |
| ePK.SD.ECKA | Ephemeral public key of the SD used for key agreement | see below | Mandatory |
| eSK.OCE.ECKA | Ephemeral private key of the OCE used for key agreement | see below | Mandatory |
| ePK.OCE.ECKA | Ephemeral public key of the OCE used for key agreement | see below | Mandatory |
| SK.SD.ECKA | Private key of the SD used for key agreement | see below | Mandatory |
| PK.SD.ECKA | Public key of the SD used for key agreement | see below | Mandatory |
| CERT.SD.ECKA | Certificate containing the public key of the SD used for key agreement, signed by the CA-KLCC | see below | Mandatory |
| SK.OCE.ECKA | Private key of the OCE used for key agreement | see below | SCP11a only |
| PK.OCE.ECKA | Public key of the OCE used for key agreement | see below | SCP11a only |
| CERT.OCE.ECKA | Certificate containing the public key of the OCE used for key agreement, signed by the CA-KLOC | see below | SCP11a only |
| SK.CA-KLOC.ECDSA | Private key of the CA-KLOC used for signing certificates | see below | SCP11a only |
| PK.CA-KLOC.ECDSA | Public key of the CA-KLOC used for verifying certificates | see below | SCP11a only |

All ECC keys shall reference the same curve parameters. Thus all keys have the same length. The curve parameters shall be available on the SD prior to any SCP11 related operation.

It is recommended to use one of the standardized curves from [Amd E].

SK.SD.ECKA and PK.CA-KLOC.ECDSA are keys stored in the SD supporting SCP11, each with its own unique combination of Key Identifier and Key Version Number. Several of these keys may be stored in an SD.

The following Key Identifiers (KID) shall be used:

- KID '10' for PK.CA-KLOC.ECDSA
- KID '11' for SK.SD.ECKA used for SCP11a
- KID '12' for the optional static Key-DEK used with SCP11a
- KID '13' for SK.SD.ECKA used for SCP11b
- KID '14' for the optional static Key-DEK used with SCP11b

***Note:*** *Even if KID values are fixed, they are also provided in the commands defined in Chapter 6. This allows reusing the commands for other purposes in the future.*

***Note:*** *Assignments of Key Version Numbers (KVN) may be defined in configurations.*

A related pair of SK.SD.ECKA and Key-DEK shall have the same KVN.

CERT.SD.ECKA is a data object stored in the SD supporting SCP11, referencing the Key Identifier and Key Version Number of the associated SK.SD.ECKA (see section 6.6). Each SK.SD.ECKA must have one associated CERT.SD.ECKA.

When contained in a command or a response, static or ephemeral public keys shall be formatted using uncompressed encoding as specified in section 3.1.1 of [TR 03111], with most significant byte coming first (hence the value shall start with the coding identifier byte '04'). Thus each key value field will have a fixed length of twice the order length plus one. For ephemeral public keys, this key value field is the data field of the TLV with tag '5F49'.

***Note:*** *The ephemeral private keys and the shared secrets ShSs, ShSe, and ShS are as sensitive as the static private keys and need to be protected accordingly.*

## 5.2 AES Keys

**Table 5-2: Security Domain Secure Channel Keys**

| Key | Usage | Length | Remark |
|---|---|---|---|
| Data Encryption Key (Key-DEK) | Sensitive Data Encryption and Decryption (AES) | see below | Optional |
| Session Data Encryption Key (S-DEK) | Sensitive Data Encryption and Decryption (AES) | see below | Conditional / Dynamically |
| Secure Channel Session Encryption Key (S-ENC) | Used for data confidentiality | see below | Dynamically |
| Secure Channel Session Message Authentication Code Key for Command (S-MAC) | Used for data and protocol integrity | see below | Dynamically |
| Secure Channel Session Message Authentication Code Key for Response (S-RMAC) | Used for data and protocol integrity | see below | Dynamically |

See section 5.1 for KID values for the Key-DEK.

If the static Key-DEK is not present in the SD, a session DEK (S-DEK) will be generated together with the other session keys. It will be used for sensitive data encryption and decryption during the secure channel session instead of the Key-DEK.

The recommended length of these AES keys depends on the length of the ECC keys according to the following table:

**Table 5-3: Recommended Length of AES Keys**

| ECC Key Length in Bits | Recommended Length of AES Keys in Bits |
|---|---|
| 256-383 | 128 |
| 384-511 | 192 |
| 512+ | 256 |

***Note:*** *To provide a balanced security, it is strongly recommended to implement this pairing. However, an implementation may also choose to tolerate other combinations. The security implications have to be considered carefully. For example, if ECC 256 is used to establish AES256 keys, these AES keys cannot be considered to provide their full strength.*

***Note:*** *Although SCP11 uses the secure messaging mechanisms of SCP03, an SD may support SCP03 with static keys as specified in [Amd D] independently. Support for SCP11 does not imply support for SCP03 nor affect any configuration settings for SCP03.*

## 5.3 Cryptographic Usage

### 5.3.1 AES Session Keys

AES session keys shall be generated every time a Secure Channel is initiated and are used for secure messaging on subsequent commands.

Session keys are generated to ensure that a different set of keys is used for each Secure Channel session.

### 5.3.2 Secure Messaging

Secure Messaging as defined in [Amd D] shall be applied to all commands following a successful MUTUAL or INTERNAL AUTHENTICATE command.

Only two security levels for Secure Messaging are defined in this specification; the security level is set in the key usage qualifier data object of the MUTUAL or INTERNAL AUTHENTICATE command:

- C-MAC and R-MAC only
- C-DECRYPTION, R-ENCRYPTION, C-MAC, and R-MAC

The MAC chaining value of the first APDU command after the MUTUAL or INTERNAL AUTHENTICATE command shall be set to the value of the receipt returned by the SD in the MUTUAL or INTERNAL AUTHENTICATE response.

> ***Note:*** *When using SCP03, the first MAC is calculated on the EXTERNAL AUTHENTICATE command and the following commands are bound to the secure channel initiation via the MAC chaining. As the equivalent in SCP11 for this first MAC is the receipt, it is used as the first MAC chaining value to bind the following commands to the SCP11 secure channel initiation.*

### 5.3.3 Key Access Conditions

The Key Access Conditions as defined in [GPCS] shall be supported by the SD for SK.SD.ECKA and Key-DEK.

Its value shall be interpreted as follows:

- Setting up a Secure Channel with an SK.SD.ECKA having Key Access Conditions set to '00' shall always be accepted.
- Setting up a Secure Channel with an SK.SD.ECKA having Key Access Conditions set to '01' shall only be accepted if the SD holding the keys is the selected or the targeted application.
- Setting up a Secure Channel with an SK.SD.ECKA having Key Access Conditions set to '02' shall only be accepted if an application associated to the SD holding the keys is the selected or the targeted application. If the associated application is an SD, additional requirements may apply (e.g. the SD not having a key set of its own).

  ***Note:*** *The setting '02' for SK.SD.ECKA is useful to prevent a secure channel from being established by the SD and being used for application management or key update.*
- An attempt by the SD to use the Key-DEK having Key Access Conditions set to '02' shall fail. It shall be allowed for Key Access Conditions set to '00' or '01'.
- An attempt by an application (including SDs) associated to the SD to use the Key-DEK having Key Access Conditions set to '01' shall fail. It shall be allowed for Key Access Conditions set to '00' or '02'.

# 6 Commands

The following table presents the new commands involved in Secure Channel Initiation and in SD Personalization when SCP11a/b is used:

**Table 6-1: SCP11 Command Support**

| Command | Used By |
|---|---|
| GET DATA (ECKA Certificate) | SCP11a and b |
| PERFORM SECURITY OPERATION | SCP11a |
| MUTUAL AUTHENTICATE | SCP11a |
| INTERNAL AUTHENTICATE | SCP11b |
| STORE DATA (ECKA Certificate) | SCP11a and b |
| STORE DATA (Whitelist) | SCP11a |

***Note:*** *STORE DATA for the key establishment scenarios in Card Specification Amendments A and E use data structures with the same CRT. However, these can be clearly distinguished as the scenarios use DGI format whereas SCP11 uses TLV format.*

## 6.1 General Coding Rules

### 6.1.1 SCP Identifier and Parameters

The value field of SCP identifier and parameters shall be coded as follows:

The SCP identifier (byte 1) shall be set to '11'.

The SCP parameters (byte 2) are defined as follows:

**Table 6-2: Parameters for SCP11**

| b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | Description |
|---|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | X | 0: Indicates SCP11b<br>1: Indicates SCP11a |
| – | – | – | – | – | X | – | – | 0: Do not include Host and Card ID in key derivation process<br>1: Include Host and Card ID in key derivation process |
| X | X | X | X | X | – | X | – | RFU (0) |

*Note: The use of b3 is aligned with the coding defined in [Amd E].*

## 6.2 GET DATA (ECKA Certificate) Command

The GET DATA command is defined in section 11.3 of [GPCS].

It is used by the OCE to retrieve a CERT.SD.ECKA from the SD.

The SD shall support bit b8 of the class byte set to 1 – support for bit b8 set to 0 is optional; the instruction code shall be set to 'CA'.

The parameters P1 and P2 shall be set to 'BF 21'.

The data field of the command message shall be coded according to the following table:

**Table 6-3: Data Field of GET DATA (ECKA Certificate) Command**

| Tag | Length | Value Description | | | MOC |
|---|---|---|---|---|---|
| 'A6' | 4 | Control Reference Template (Key Agreement) | | | M |
| | | **Tag** | **Length** | **Value Description** | **MOC** |
| | | '83' | 2 | byte 1: Key Identifier<br>byte 2: Key Version Number | M |

The SD shall return the CERT.SD.ECKA linked to the private key SK.SD.ECKA which is referenced by the CRT, encapsulated according to the following table:

**Table 6-4: Data Field of GET DATA (ECKA Certificate) Response**

| Tag | Length | Value Description | | | MOC |
|---|---|---|---|---|---|
| 'BF 21' | Var | SCP11 certificate store | | | M |
| | | **Tag** | **Length** | **Value Description** | **MOC** |
| | | '7F 21' | Var | CERT.SD.ECKA | M |

The format of the CERT.SD.ECKA is identical to the format defined for CERT.CASD.ECKA in [Amd E]. Tag '42' identifies the owner of the SD; tag '45' identifies the Security Domain Image Number.

The OCE shall verify at least the following:

- The signature of the CERT.SD.ECKA, using the PK.CA-KLCC.ECDSA
- The Expiration Date, to ensure that the certificate is still valid
- The correctness of the Key Usage

In addition, the OCE should check either a whitelist or a revocation list for the SD's certificates which should be maintained off card.

***Note:*** *The OCE can retrieve the Key Version Number and Key Identifier of the ECC keys available in the SD by retrieving the Key Information Template using a GET DATA command. A stored PK.OCE.ECKA is not included in the Key Information Template.*

## 6.3 PERFORM SECURITY OPERATION Command

### 6.3.1 Definition and Scope

The PERFORM SECURITY OPERATION command is used to send the OCE certificate to the SD. This is required as a precondition to the initiation of an SCP11a secure channel.

The command does not terminate an ongoing secure channel session.

The SD shall verify the following data of the certificate:

- The signature of the certificate, using the PK.CA-KLOC.ECDSA referenced in the command.
- If a whitelist is linked to this key (see section 6.7), the SD also verifies that the Certificate Serial Number of the OCE certificate is contained in the whitelist.
- The correct value for Key Usage.
- The structure of the public key including the existence of the referenced key parameters.

All other fields of the certificate may be ignored by the SD.

If these verifications are successful, the SD shall extract the OCE's public key and, dependent on the implementation option (see section 4.2), store it persistently or temporarily for use in (a) subsequent MUTUAL AUTHENTICATE command(s).

If the key is stored persistently, only one key shall be stored per SD; the command shall cause a previously stored public key to be replaced.

***Note:*** *An update of a PK.CA-KLOC.ECDSA does not have any effect on a stored PK.OCE.ECKA. If the OCE's static key pair is (also) no longer trusted, the PK.OCE.ECKA stored in the SD has to be updated by a separate PERFORM SECURITY OPERATION command.*

### 6.3.2 Command Message

The PERFORM SECURITY OPERATION command message shall be coded according to the following table:

**Table 6-5: PERFORM SECURITY OPERATION Command Message**

| Code | Value | Meaning |
|---|---|---|
| CLA | '80' - '87', 'C0' - 'CF', or 'E0' - 'EF' | See [GPCS] section 11.1.4. |
| INS | '2A' | PERFORM SECURITY OPERATION |
| P1 | 'xx' | Key Version Number |
| P2 | 'xx' | Key Identifier |
| Lc | 'xx' | Length of data filed |
| Data | 'xx xx...' | Certificate |
| Le | '00' | |

***Note:*** *The command can also be sent in a secure channel session (see range for CLA), which will modify the data structure (e.g. adding MACs, etc.). A subsequent MUTUAL AUTHENTICATE command will terminate this session and initiate a new secure channel session.*

#### 6.3.2.1 Reference Control Parameter P1

Reference control parameter P1 references the Key Version Number of the PK.CA-KLOC.ECDSA, which is used to verify the certificate's signature. It is coded on bits 1..7. Bit 8 is RFU and set to zero.

#### 6.3.2.2 Reference Control Parameter P2

Reference control parameter P2 references the Key Identifier of the PK.CA-KLOC.ECDSA, which is used to verify the certificate's signature. It is coded on bits 1..7. Bit 8 is RFU and set to zero.

#### 6.3.2.3 Data Field Sent in the Command Message

The data field of the command message shall be coded according to the following table:

**Table 6-6: PERFORM SECURITY OPERATION Command Data**

| Tag | Length | Value Description | | | MOC |
|---|---|---|---|---|---|
| '7F21' | Var | Certificate | | | M |
| | | **Tag** | **Length** | **Value Description** | **MOC** |
| | | '93' | 1-16 | Certificate Serial Number | M |
| | | '42' | 1-16 | CA-KLOC Identifier | M |
| | | '5F20' | 1-16 | Subject Identifier | M |
| | | '95' | 1 | Key Usage, Signature Verification | M |
| | | '5F25' | 4 | Effective Date (YYYYMMDD, BCD format) | O |
| | | '5F24' | 4 | Expiration Date (YYYYMMDD, BCD format) | M |
| | | '53' or '73' | 1-127 | Discretionary Data | O |
| | | '7F49' | Var | Public Key – for details see tables below | M |
| | | '5F37' | Var | Signature | M |

The following TLV-encoded data are signed off-card with SK.CA-KLOC.ECDSA to generate the content of tag '5F37' (signature), as described in [Amd E]:

**Table 6-7: Data Signed to Generate the OCE Certificate**

| Tag | Length | Value Description | MOC |
|---|---|---|---|
| '93' | 1-16 | Certificate Serial Number | M |
| '42' | 1-16 | CA-KLOC Identifier | M |
| '5F20' | 1-16 | Subject Identifier | M |
| '95' | 1 | Key Usage, Signature Verification | M |
| '5F25' | 4 | Effective Date (YYYYMMDD, BCD format) – if present | C |
| '5F24' | 4 | Expiration Date (YYYYMMDD, BCD format) | M |
| '53' or '73' | 1-127 | Discretionary Data – if present | C |
| '7F49' | Var | Public Key | M |

The Public Key Data Object contains an Elliptic Curves (EC) public key and the corresponding key parameter reference.

**Table 6-8: Public Key Data Object**

| Tag | Length | Value Description | | | MOC |
|---|---|---|---|---|---|
| '7F49' | Var | Public Key Data Object | | | M |
| | | **Tag** | **Length** | **Value Description** | **MOC** |
| | | 'B0' | Var | Public key – Q | M |
| | | 'F0' | 1 or 2 | Key Parameter Reference | M |

### 6.3.3 Response Message

#### 6.3.3.1 Data Field Returned in the Response Message

The data field of the response message shall not be present.

#### 6.3.3.2 Processing State Returned in the Response Message

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

This command may either return a general error condition as listed in [GPCS] section 11.1.3 or one of the following error conditions.

**Table 6-9: PERFORM SECURITY OPERATION Error Conditions**

| SW1 | SW2 | Meaning |
|---|---|---|
| '66' | '00' | Verification of the certificate failed |
| '66' | '40' | Certificate not in whitelist |
| '6A' | '80' | Incorrect values in command data |
| '6A' | '88' | Referenced PK.CA-KLOC.ECSDA not found |

## 6.4 MUTUAL AUTHENTICATE Command

### 6.4.1 Definition and Scope

The MUTUAL AUTHENTICATE command is used to send the ephemeral public key of the OCE to the SD, to trigger the key establishment, to provide card authentication information to the OCE, and to determine the level of security required for all subsequent commands.

The MUTUAL AUTHENTICATE command terminates an ongoing secure channel session (whichever secure channel protocol is currently used) and if the command is successful, initiates a new secure channel session.

If no PK.OCE.ECKA has been provided to the SD earlier in a PERFORM SECURITY OPERATION command, the MUTUAL AUTHENTICATE command shall fail with error condition “conditions of use not satisfied”.

If the PK.OCE.ECKA was not provided immediately before the MUTUAL AUTHENTICATE command, the OCE can check that the SD used the correct PK.OCE.ECKA by verifying the receipt generated by the SD.

### 6.4.2 Command Message

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

**Table 6-10: MUTUAL AUTHENTICATE Command Message**

| Code | Value | Meaning |
|---|---|---|
| CLA | '80' - '83' or 'C0' - 'CF' | See [GPCS] section 11.1.4. |
| INS | '82' | MUTUAL AUTHENTICATE |
| P1 | 'xx' | Key Version Number |
| P2 | 'xx' | Key Identifier |
| Lc | 'xx' | Length data field |
| Data | 'xx xx…' | Data for key establishment |
| Le | '00' | |

***Note:*** *INS for MUTUAL AUTHENTICATE is the same as for EXTENAL AUTHENTICATE used in SCP02 and SCP03. However, the P2 value, which is set to '00' for SCP02 and SCP03, is always different for SCP11.*

#### 6.4.2.1 Reference Control Parameter P1

Reference control parameter P1 references the Key Version Number of the SK.SD.ECKA. It is coded on bits 1..7. Bit 8 is RFU and set to zero.

#### 6.4.2.2 Reference Control Parameter P2

Reference control parameter P2 references the Key Identifier of the SK.SD.ECKA. It is coded on bits 1..7. Bit 8 is RFU and set to zero.

#### 6.4.2.3 Data Field Sent in the Command Message

The data field of the command message shall be coded according to the following table:

**Table 6-11: MUTUAL AUTHENTICATE Data Field**

| Tag | Length | Value Description | | | MOC |
|---|---|---|---|---|---|
| 'A6' | Var | Control Reference Template (Key Agreement) | | | M |
| | | **Tag** | **Length** | **Value Description** | **MOC** |
| | | '90' | 2 | SCP identifier and parameters (see section 6.1.1) | M |
| | | '95' | 1 | Key Usage Qualifier<br>• '34' (secure messaging with MAC only) or<br>• '3C' (secure messaging with MAC and ENCRYPTION)<br>(See [GPCS] Table 11-17) | M |
| | | '80' | 1 | Key Type according to [GPCS] Table 11-16<br>• '88' (AES) | M |
| | | '81' | 1 | Key Length (in bytes) | M |
| | | '84' | 1-n | HostID (shall only be present if SCP parameter b3 is set) | C |
| '5F49' | Var | ePK.OCE.ECKA | | | M |

The SD shall verify the values provided for SCP identifier, SCP parameters, key usage qualifier, and key type.

If mandated by the security policy, key length shall be checked according to the recommendations defined in section 5.2.

If bit 3 of the SCP parameters is set (“Include Host and Card ID in key derivation process”) and tag '84' (Host ID) is not present within tag 'A6', then an error shall be returned. Similarly, if bit 3 is not set and tag '84' (Host ID) is present within tag 'A6', then an error shall be returned.

The SD shall generate an ephemeral key pair eSK.SD.ECKA and ePK.SD.ECKA.

The SD shall use PK.OCE.ECKA and SK.SD.ECKA to generate the shared secret ShSs according to section 3.1.1.

The SD shall use ePK.OCE.ECKA and eSK.SD.ECKA to generate the shared secret ShSe according to section 3.1.1.

The SD shall concatenate ShSe and ShSs to form the shared secret ShS which constitutes the input for the Key Derivation process.

The concatenation of the following values shall be used for *SharedInfo* as input for the Key Derivation process:

- Key usage qualifier (1 byte)
- Key type (1 byte)
- Key length (1 byte)
- If Host and Card ID are requested: HostID-LV, SIN-LV, and SDIN-LV

***Note:*** *The presence of unique host (off card entity) and card identifiers is required in [NIST 800-56A].*

HostID-LV is the length and the value field of the HostID given in the command data.

SIN-LV is the length and the value field of the Security Domain Provider Identification Number of the SD (see [GPCS]).

SDIN-LV is the length and the value field of the Security Domain Image Number of the SD (see [GPCS]).

SHA-256 shall be used for the key derivation to calculate *KeyData* of sufficient length, which is then assigned to keys as defined below.

***Note:*** *SHA-256 is considered strong enough even for AES-256 keys, and the output size aligns nicely with most key lengths.*

In addition to the session keys, a receipt key is used to calculate the receipt to be included in the response to the MUTUAL AUTHENTICATE command. The type and length of the receipt key is the same as for the session keys.

The *KeyData* generated as defined in section 3.1.2 shall be assigned to the keys as follows (L is the key length):

**Table 6-12: *KeyData* Assignment**

| ***KeyData*** | **Key** |
|---|---|
| 1 to L | Receipt key |
| L+1 to 2L | S-ENC |
| 2L+1 to 3L | S-MAC |
| 3L+1 to 4L | S-RMAC |
| 4L+1 to 5L | S-DEK (if no Key-DEK is present) |

Finally, the SD shall generate a receipt (using the receipt key and the MAC algorithm used in the secure channel) by calculating a MAC across the data described in Table 6-13. The receipt key shall be deleted after calculating the receipt.

**Table 6-13: Input Data for Receipt Calculation**

| **Tag** | **Length** | **Data Element** | **Presence** |
|---|---|---|---|
| 'A6' | Variable | CRT TLV with all sub TLVs as provided in the MUTUAL AUTHENTICATE command | Mandatory |
| '5F49' | Variable | ePK.OCE.ECKA | Mandatory |
| '5F49' | Variable | ePK.SD.ECKA | Mandatory |

### 6.4.3 Response Message

#### 6.4.3.1 Data Field Returned in the Response Message

The data field of the response message shall contain the following data objects:

**Table 6-14: MUTUAL AUTHENTICATE Response Data**

| Tag | Length | Value Description | MOC |
|---|---|---|---|
| '5F49' | Variable | ePK.SD.ECKA | M |
| '86' | 16 | Receipt | M |

#### 6.4.3.2 Processing State Returned in the Response Message

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

This command may either return a general error condition as listed in [GPCS] section 11.1.3 or one of the following error conditions.

**Table 6-15: MUTUAL AUTHENTICATE Error Conditions**

| SW1 | SW2 | Meaning |
|---|---|---|
| '6A' | '80' | Incorrect values in command data |
| '6A' | '88' | One of the following referenced data elements is not found: SK.SD.ECSDA / PK.OCE.ECKA / SIN / SDIN |

## 6.5 INTERNAL AUTHENTICATE Command

### 6.5.1 Definition and Scope

The INTERNAL AUTHENTICATE command is used to trigger the key establishment, to provide card authentication information to the OCE, and to determine the level of security required for all subsequent commands.

The INTERNAL AUTHENTICATE command terminates an ongoing secure channel session (whichever secure channel protocol is currently used) and if the command is successful, initiates a new secure channel session.

### 6.5.2 Command Message

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

**Table 6-16: INTERNAL AUTHENTICATE Command Message**

| Code | Value | Meaning |
|---|---|---|
| CLA | '80' - '83' or 'C0' - 'CF' | See [GPCS] section 11.1.4. |
| INS | '88' | INTERNAL AUTHENTICATE |
| P1 | 'xx' | Key Version Number |
| P2 | 'xx' | Key Identifier |
| Lc | 'xx' | Length data field |
| Data | 'xx xx…' | Data for key establishment |
| Le | '00' | |

#### 6.5.2.1 Reference Control Parameter P1

Reference control parameter P1 references the Key Version Number of the SK.SD.ECKA. It is coded on bits 1..7. Bit 8 is RFU and set to zero.

#### 6.5.2.2 Reference Control Parameter P2

Reference control parameter P2 references the Key Identifier of the SK.SD.ECKA. It is coded on bits 1..7. Bit 8 is RFU and set to zero.

#### 6.5.2.3 Data Field Sent in the Command Message

The data field of the command message shall be coded according to the following table:

**Table 6-17: INTERNAL AUTHENTICATE Data Field**

| Tag | Length | Value Description | | | MOC |
|---|---|---|---|---|---|
| 'A6' | Var | Control Reference Template (Key Agreement) | | | M |
| | | **Tag** | **Length** | **Value Description** | **MOC** |
| | | '90' | 2 | SCP identifier and parameters (see section 6.1.1) | M |
| | | '95' | 1 | Key Usage Qualifier<br>• '34' (secure messaging with MAC only) or<br>• '3C' (secure messaging with MAC and ENCRYPTION)<br>(See [GPCS] Table 11-17) | M |
| | | '80' | 1 | Key Type according to [GPCS] Table 11-16<br>• '88' (AES) | M |
| | | '81' | 1 | Key Length (in bytes) | M |
| | | '84' | 1-n | HostID (shall only be present if SCP parameter b3 is set) | C |
| '5F49' | Var | ePK.OCE.ECKA | | | M |

Processing shall be done as defined for MUTUAL AUTHENTICATE in section 6.4.2.3, with the following modifications:

- Bit 4 of the SCP parameters is different (see section 6.1.1).
- Instead of PK.OCE.ECKA, SD shall use ePK.OCE.ECKA when generating the shared secret ShSs; i.e. ePK.OCE.ECKA is used in the calculation of both ShSs and ShSe.

### 6.5.3 Response Message

#### 6.5.3.1 Data Field Returned in the Response Message

The data field of the response message shall contain the following data objects:

**Table 6-18: INTERNAL AUTHENTICATE Response Data**

| Tag | Length | Value Description | MOC |
|---|---|---|---|
| '5F49' | Variable | ePK.SD.ECKA | M |
| '86' | 16 | Receipt | M |

#### 6.5.3.2 Processing State Returned in the Response Message

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

This command may either return a general error condition as listed in [GPCS] section 11.1.3 or one of the following error conditions.

**Table 6-19: INTERNAL AUTHENTICATE Error Conditions**

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

## 6.6 STORE DATA (ECKA Certificate) Command

The STORE DATA command is defined in section 11.11 of [GPCS]. BER-TLV format shall be used for the command data.

To store or replace a certificate linked to a private key, the data field of the command message shall contain two BER-TLVs as defined in the following table:

**Table 6-20: Data Field of STORE DATA (ECKA Certificate) Command**

<table>
<tr><th>Tag</th><th>Length</th><th colspan="3">Value Description</th><th>MOC</th></tr>
<tr><td rowspan="3">'A6'</td><td rowspan="3">4</td><td colspan="3">Control Reference Template (Key Agreement)</td><td>M</td></tr>
<tr><td>Tag</td><td>Length</td><td>Value Description</td><td>MOC</td></tr>
<tr><td>'83'</td><td>2</td><td>byte 1: Key Identifier<br>byte 2: Key Version Number</td><td>M</td></tr>
<tr><td>'BF21'</td><td>Var</td><td colspan="3">SCP11 certificate store (see section 6.2)</td><td>M</td></tr>
</table>

After successful execution of the command, the SCP11 certificate store is linked to the private key SK.SD.ECKA referenced by the CRT.

When the SK.SD.ECKA is deleted or replaced, the SCP11 certificate store shall be automatically deleted by the SD.

If the referenced SK.SD.ECKA does not exist in the SD, the command shall be rejected with error condition '6A88'.

## 6.7 STORE DATA (Whitelist) Command

The STORE DATA command is defined in section 11.11 of [GPCS]. BER-TLV format shall be used for the command data.

To store or replace a whitelist linked to a PK.CA-KLOC.ECDSA, the data field of the command message shall contain two BER-TLVs as defined in the following table:

**Table 6-21: Data Field of STORE DATA (Whitelist) Command**

| Tag | Length | Value Description | | | MOC |
|---|---|---|---|---|---|
| 'A6' | 4 | Control Reference Template (Key Agreement) | | | M |
| | | **Tag** | **Length** | **Value Description** | **MOC** |
| | | '83' | 2 | byte 1: Key Identifier<br>byte 2: Key Version Number | M |
| '70' | Var | Whitelist | | | M |
| | | **Tag** | **Length** | **Value Description** | **MOC** |
| | | '93' | 1-16 | Certificate Serial Number | O |
| | | '93' | 1-16 | Certificate Serial Number | O |
| | | … | … | … | … |
| | | '93' | 1-16 | Certificate Serial Number | O |

After successful execution of the command, the whitelist of the command replaces any previously stored whitelist linked to the PK.CA-KLOC.ECDSA referenced by the CRT. To remove a whitelist, the whitelist TLV of the command shall have a length of zero.

When a whitelist is newly stored or replaced, a stored PK.OCE.ECKA shall be deleted by the SD.

When a PK.CA-KLOC.ECDSA is deleted, a whitelist linked to it shall also be deleted by the SD. An update of a PK.CA-KLOC.ECDSA has no impact on a linked whitelist.

# Annex A OCE Authentication for SCP11b

SCP11b provides authentication of the card to the Off Card Entity (OCE) only. Mechanisms for authentication of the OCE to the card are out of scope of the secure channel protocol and have to be provided by applications that are using SCP11b.

This annex provides an example for a mechanism based on PIN verification which could be used for this purpose.

## A.1 OCE Providing PIN Verification

A weak authentication mechanism can be provided by the OCE by sending a PIN code to the SD, which was entered by the user at the OCE's user interface. Strictly speaking, this authenticates the user. However, if the user is instructed to enter the PIN only on the user interface of the device hosting the OCE's endpoint of the secure channel, this indirectly also authenticates the OCE.

This approach may be used in certain cases where the OCE is (a Trusted Application in) a Trusted Execution Environment, providing a Trusted User Interface (see the GlobalPlatform Trusted User Interface API, [TUI]).

The detailed interaction between the card and the device is out of scope of this specification.

This annex just provides the typical command used by the application in such a scenario once SCP11b is established: VERIFY PIN.

The VERIFY PIN command message in the secure channel is coded according to the following table:

**Table A-1: VERIFY PIN Command Message**

| Code | Value | Meaning |
|---|---|---|
| CLA | '84' - '87', or 'E0' - 'EF' | Please refer to [GPCS] section 11.1.4 |
| INS | '20' | VERIFY PIN |
| P1 | '00' | Reference control parameter P1:<br>Normal operation |
| P2 | '00' | Reference control parameter P2:<br>No information given |
| Lc | 'XX' | Length of data field |
| Data | 'xx xx…' | PIN value |
| Le | | Not present |

### A.1.1 Data Field Sent in the Command Message

The data field contains the UTF-8 encoded PIN entered by the user.

### A.1.2 Processing State Returned in the Response Message

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

This command may either return a general error condition as listed in [GPCS] section 11.1.3 or one of the following error conditions.

**Table A-2: VERIFY PIN Error Conditions**

| SW1 | SW2 | Meaning |
|---|---|---|
| '63' | 'CX' | Authentication failed, X retries allowed |
| '69' | '83' | Authentication failed, PIN blocked |