

GlobalPlatform Technology
Secure Channel Protocol '11'
Card Specification v2.3 – Amendment F
Version 1.2.1.3 (target 1.3)

Public Review

February 2021

Document Reference: GPC_SPE_093

Copyright © 2014-2021 GlobalPlatform, Inc. All Rights Reserved.

Recipients of this document are invited to submit, with their comments, notification of any relevant patents or other intellectual property rights (collectively, "IPR") of which they may be aware which might be necessarily infringed by the implementation of the specification or other work product set forth in this document, and to provide supporting documentation. The technology provided or described herein is subject to updates, revisions, and extensions by GlobalPlatform. Use of this information is governed by the GlobalPlatform license agreement and any use inconsistent with that agreement is strictly prohibited.

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

1	Introduction	8
1.1	Audience	8
1.2	IPR Disclaimer	8
1.3	References	8
1.4	Terminology and Definitions	9
1.5	Abbreviations and Notations	9
1.6	Revision History	12
2	Secure Channel Protocol '11'	13
2.1	Scope of the Document.....	13
2.2	Use Cases and Requirements	13
3	Specification Amendments.....	14
3.1	Algorithms	14
3.1.1	ECKA.....	14
3.1.2	Key Derivation.....	14
3.2	Controlling Authority Roles	15
3.3	Certificate Verification	16
4	Secure Channel Protocol Usage	18
4.1	Protocol Overview	18
4.2	Secure Communication Configuration	21
4.3	Authentication	22
4.4	Message Integrity and Data Confidentiality	22
4.5	Forward Secrecy.....	23
4.6	Session Replay	23
4.7	API and Security Level.....	24
4.8	Protocol Rules.....	25
5	Cryptographic Keys	26
5.1	ECC Keys.....	26
5.2	AES Keys	29
5.3	Cryptographic Usage	30
5.3.1	AES Session Keys	30
5.3.2	Secure Messaging	30
5.3.3	Key Access Conditions	31
6	Commands.....	32
6.1	General Coding Rules.....	32
6.1.1	SCP Identifier and Parameters	32
6.2	GET DATA (ECKA Certificate) Command	33
6.2.1	Response Message	34
6.3	GET DATA (CA-KLOC KID-KVN) Command	35
6.4	PERFORM SECURITY OPERATION Command.....	36
6.4.1	Definition and Scope	36
6.4.2	Command Message	37
6.4.2.1	Reference Control Parameter P1.....	37
6.4.2.2	Reference Control Parameter P2.....	38
6.4.2.3	Example of Using P1P2	38
6.4.2.4	Data Field Sent in the Command Message	39
6.4.3	Response Message	41
6.4.3.1	Data Field Returned in the Response Message	41

6.4.3.2	Processing State Returned in the Response Message	41
6.5	MUTUAL AUTHENTICATE Command	42
6.5.1	Definition and Scope	42
6.5.2	Command Message	42
6.5.2.1	Reference Control Parameter P1	42
6.5.2.2	Reference Control Parameter P2	42
6.5.2.3	Data Field Sent in the Command Message	43
6.5.3	Response Message	45
6.5.3.1	Data Field Returned in the Response Message	45
6.5.3.2	Processing State Returned in the Response Message	45
6.6	INTERNAL AUTHENTICATE Command	47
6.6.1	Definition and Scope	47
6.6.2	Command Message	47
6.6.2.1	Reference Control Parameter P1	47
6.6.2.2	Reference Control Parameter P2	47
6.6.2.3	Data Field Sent in the Command Message	48
6.6.3	Response Message	48
6.6.3.1	Data Field Returned in the Response Message	48
6.6.3.2	Processing State Returned in the Response Message	49
6.7	STORE DATA (ECKA Certificate) Command	50
6.8	STORE DATA (Whitelist) Command	51
6.9	STORE DATA (CA-KLOC Identifier) Command	53
Annex A	OCE Authentication for SCP11b	54
A.1	OCE Providing PIN Verification	54
A.1.1	Data Field Sent in the Command Message	54
A.1.2	Processing State Returned in the Response Message	55
Annex B	Usage of SCP11c	56
B.1	Authentication Level	56
B.2	Limitations on Commands Received through SCP11c	57
B.3	Detection and Removal of Pending Card Contents	61
B.4	PERFORM TRANSACTION OPERATION Command	62
B.4.1	Definition and Scope	62
B.4.2	Command Message	62
B.4.3	Response Message	62
B.5	Self-Deletion via DELETE	63
B.6	Cumulative DELETE by Parent SD	64

Figures

Figure 3-1: Single Certificate (No Chaining).....	16
Figure 3-2: Certificate Chain.....	17
Figure 4-1: Initial Certificate Retrieval	18
Figure 4-2: SCP11a Protocol Overview	19
Figure 4-3: SCP11b Protocol Overview	19
Figure 4-4: SCP11c Protocol Overview.....	20

Tables

Table 1-1: Normative References.....	8
Table 1-2: Informative References	9
Table 1-3: Terminology and Definitions.....	9
Table 1-4: Abbreviations and Notations	9
Table 1-5: Revision History	12
Table 4-1: Values of Parameter “i”	21
Table 5-1: ECC Keys.....	26
Table 5-2: Security Domain Secure Channel Keys	29
Table 5-3: Recommended Length of AES Keys.....	29
Table 6-1: SCP11 Command Support.....	32
Table 6-2: Parameters for SCP11	32
Table 6-3: Data Field of GET DATA (ECKA Certificate(s)) Command	33
Table 6-4: Data Field of GET DATA (ECKA Certificate(s)) Response.....	33
Table 6-5: Data Field of GET DATA (CA-KLOC KID-KVN) Command	35
Table 6-6: Data Field of GET DATA (CA-KLOC KID-KVN) Response	35
Table 6-7: PERFORM SECURITY OPERATION Command Message	37
Table 6-8: Values of Reference Control Parameter P1.....	37
Table 6-9: Values of Reference Control Parameter P2.....	38
Table 6-10: Example of Using P1P2	38
Table 6-11: PERFORM SECURITY OPERATION Command Data	39
Table 6-12: Certificate Format.....	39
Table 6-13: Data Signed to Generate the Certificate	40
Table 6-14: Public Key Data Object	40
Table 6-15: PERFORM SECURITY OPERATION Error Conditions.....	41
Table 6-16: MUTUAL AUTHENTICATE Command Message	42
Table 6-17: MUTUAL AUTHENTICATE Data Field	43
Table 6-18: <i>KeyData</i> Assignment.....	45
Table 6-19: Input Data for Receipt Calculation	45
Table 6-20: MUTUAL AUTHENTICATE Response Data	45
Table 6-21: MUTUAL AUTHENTICATE Error Conditions.....	46
Table 6-22: INTERNAL AUTHENTICATE Command Message	47
Table 6-23: INTERNAL AUTHENTICATE Data Field	48
Table 6-24: INTERNAL AUTHENTICATE Response Data	48
Table 6-25: INTERNAL AUTHENTICATE Error Conditions.....	49

Table 6-26: Data Field of STORE DATA (ECKA Certificate) Command.....	50
Table 6-27: Data Field of STORE DATA (Whitelist) Command	51
Table 6-28: Data Field of STORE DATA (CA-KLOC Identifier) Command.....	53
Table 6-29: STORE DATA Error Conditions	53
Table A-1: VERIFY PIN Command Message.....	54
Table A-2: VERIFY PIN Error Conditions	55
Table B-1: Authentication Levels in SCP11c.....	56
Table B-2: Contents of Tag 'BF20' in OCE Certificate	57
Table B-3: Possible Authorizations per APDU Command	60
Table B-4: PERFORM TRANSACTION OPERATION Command Message	62
Table B-5: PERFORM TRANSACTION OPERATION Error Conditions.....	62
Table B-6: Values for Reference Control Parameter P2	63

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://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

Table 1-1: Normative References

Standard / Specification	Description	Ref
GPCS	GlobalPlatform Technology Card Specification v2.3	[GPCS]
GPCS Amendment A	GlobalPlatform Technology Confidential Card Content Management, Card Specification v2.3 – Amendment A, v1.1	[Amd A]
GPCS Amendment C	GlobalPlatform Technology Contactless Services, Card Specification v2.3 – Amendment C, v1.2	[Amd C]
GPCS Amendment D	GlobalPlatform Technology Secure Channel Protocol 03, Card Specification v2.3 – Amendment D, v1.2	[Amd D]
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
TEE Trusted User Interface API	GlobalPlatform Device Technology, TEE Trusted User Interface API, v1.0	[TEE TUI]

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
Card Group ID	A common identifier for a group of cards sharing the same CERT.SD.ECKA
Controlling Authority	In the context of this specification, an authority delivering certificates (root of a certification chain)
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
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
C-ENC	Command Encryption
CERT.KA-KLCC.ECDSA	Certificate containing the public key of a KA-KLCC used for verifying certificates (CERT.SD.ECKA or another intermediate CERT.KA-KLCC.ECDSA)
CERT.KA-KLOC.ECDSA	Certificate containing the public key of a KA-KLOC used for verifying certificates (CERT.OCE.ECKA or another intermediate CERT.KA-KLOC.ECDSA)

Abbreviation / Notation	Meaning
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
CGM	Cumulative Granted Memory
CLA	CLAss byte of command message
C-MAC	Command MAC
CRT	Control Reference Template
CSN	Certificate Serial Number
DGI	Data Grouping Identifier
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECKA	Elliptic Curve Key Agreement
ELF	Executable Load File
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
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
KA	Intermediate Key Authority used by CA to build certification chains
KA-KLCC	Intermediate Key Authority for Confidential Key Loading Card Certificates
KA-KLOC	Intermediate Key Authority for Confidential Key Loading OCE Certificates
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

Abbreviation / Notation	Meaning
PK.CA-KLOC.ECDSA	Public key of the CA-KLOC used for verifying certificates
PK.KA-KLCC.ECDSA	Public key of the KA-KLCC used for verifying certificates
PK.KA-KLOC.ECDSA	Public key of the KA-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
RFU	Reserved for Future Use
R-MAC	Response MAC
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 shared secrets, depending on SCP11 version a/b/c)
ShSee	Shared Secret calculated from two ephemeral keys
ShSes	Shared Secret calculated from one ephemeral (OCE) and one static (SD) key
ShSse	Shared Secret calculated from one static (OCE) and one ephemeral (SD) key
ShSss	Shared Secret calculated from two static keys
SK.CA-KLCC.ECDSA	Private key of the CA-KLCC used for signing certificates
SK.CA-KLOC.ECDSA	Private key of the CA-KLOC used for signing certificates
SK.KA-KLCC.ECDSA	Private key of the KA-KLCC used for signing certificates
SK.KA-KLOC.ECDSA	Private key of the KA-KLOC 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
TLV	Tag Length Value
UTF-8	Unicode Transformation Format – 8-bit
Var	Variable

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
May 2015	1.0	Initial Public Release
September 2017	1.1	<p>Public Release</p> <p>Added support for certificate chains down to CERT.OCE.ECKA and down to CERT.SD.ECKA.</p> <p>The root certificate authorities are still known as CA-KLOC and CA-KLCC respectively, and intermediate certificate authorities are known as KA-KLOC and KA-KLCC (i.e. Key Authorities). The GET DATA (ECKA Certificate) command can now return a chain of (on-card) certificates (i.e. Certificate Store). The PERFORM SECURITY OPERATION command can now present a chain of (off-card) certificates. See sections 3.3, 6.2, 6.4, and 6.7.</p> <p>Added support for storing several PK.CA-KLOC.ECDSA, each one corresponding to a specific CA-KLOC Identifier. See sections 5.1, 6.3, and 6.9.</p>
July 2018	1.2	<p>Public Release</p> <p>Added SCP11c, which allows the offline creation of secured command sequences (scripts) for the SE.</p> <p>If SEs are personalized with group keys, the same script can be used on a group of SEs.</p> <p>SCP11c also specifies a transaction mechanism which allows a rollback in case of failures.</p>
March 2019	1.2.1	<p>Maintenance Release</p> <p>Clarified and fixed a number of errors in Annex B (Usage of SCP11c).</p> <p>Miscellaneous clarifications about SCP11c all across the document.</p> <p>Added warning notice in section 4.3 about using certificate whitelists as a revocation mechanism.</p> <p>Clarifications + additional rules in section 4.6 (Session Replay).</p> <p>More security levels available for SCP11c (section 4.7 and 6.5).</p> <p>Clarifications for certificate chaining in sections 6.2, 6.4, and 6.7.</p> <p>Other miscellaneous clarifications.</p>
November 2020	1.2.1.2	<p>Member Review</p> <p>Alignment with changes for SCP.03 (16 byte checksum).</p> <p>Clarification on Subject ID.</p>
February 2021	1.2.1.3	Public Review
TBD	1.3	Public Release

2 Secure Channel Protocol '11'

2.1 Scope of the Document

This document specifies a 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.

SCP11 reuses cryptographic mechanisms defined in GPCS Amendment A: Confidential Card Content Management [Amd A] and in GPCS Amendment D: SCP03 [Amd D].

Three variants of the protocol are defined:

- SCP11a provides mutual authentication between the Off-Card Entity (OCE) and the card.
- SCP11b provides 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.
- SCP11c provides mutual authentication between the OCE and the card, using only static keys for the authentication of the card. This allows for offline scripting usage, authorized by the CERT.OCE.ECKA, as described in Annex B.

The secure channel can be embedded into complex use cases, e.g. installation of payment credentials on wearables, production systems, and remote provisioning of cell phone subscriptions. These use cases are out of scope of this document.

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.

SCP11c provides mutual authentication between the Off-Card Entity (OCE) and each card out of a group of cards.

- This variant can be used in scripting mode, i.e. it is possible to precompute personalization scripts for a group of cards which can be deployed later via online services or as resource data in companion apps to the entire group of cards. In this use case, the script is distributed as part of a rich environment application. Upon installation of such an application, the script is executed. No secret data is handled by the rich environment application and the cryptographic part of the OCE is run in a secure server of the OCE.
- The response (i.e. sequence of APDU responses) may be collected by the OCE, e.g. a rich environment application, and sent back to the service provider who can decrypt them using the SCP03 session keys, if required.
- If the Security Domain performs onboard key generation according to scenario #1 or scenario #3 from [Amd A], then the rich environment application stores the responses from the key generation commands and sends them back to the service provider. The service provider uses these responses to calculate the onboard generated key. This process works exactly as described in [Amd A].

3 Specification Amendments

3.1 Algorithms

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

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 BSI Technical Guideline TR-03111 [TR 03111].

ECKA used in this specification shall follow the definition for the Key Agreement Algorithm in [TR 03111]. The algorithm is executed for each variant of the protocol, as follows:

- SCP11a: With two pairs of static keys and two pairs of ephemeral keys. This scheme is described in [NIST 800-56A] as “(Cofactor) Full Unified Model, C(2e, 2s, ECC CDH)” for curves with a cofactor of 1.
- SCP11b: With two ephemeral key pairs and one static key pair. The OCE does not have a static key pair. The ephemeral key pair of the OCE is used twice.
- SCP11c: With two static key pairs and one ephemeral key pair. The SD does not create an ephemeral key pair. Instead, the static key pair of the SD is used twice. This scheme is described in [NIST 800-56A] as “One-Pass Unified Model, C(1e, 2s, ECC CDH) Scheme”.

Note: *The recommendations in [NIST 800-56A] on the handling of ephemeral keys and of intermediate results (e.g. the shared secrets $ShSee$, $ShSes$, $ShSse$, and $ShSss$) 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.

Instead of directly using the Controlling Authority (CA), it may simplify deployment to have a subordinate Key Authority (KA) to which the CA may delegate the diversification of the keys and related certificates by setting up a certificate chain. As there is no technical need for a single actor to provide both the Off-Card Entity and Security Domain roles, those roles are distinguished in this document:

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

To simplify the description of the mechanisms, we will keep using the abbreviation CA in the rest of this document, even if one or more Key Authorities are involved and a certificate chain is used.

3.3 Certificate Verification

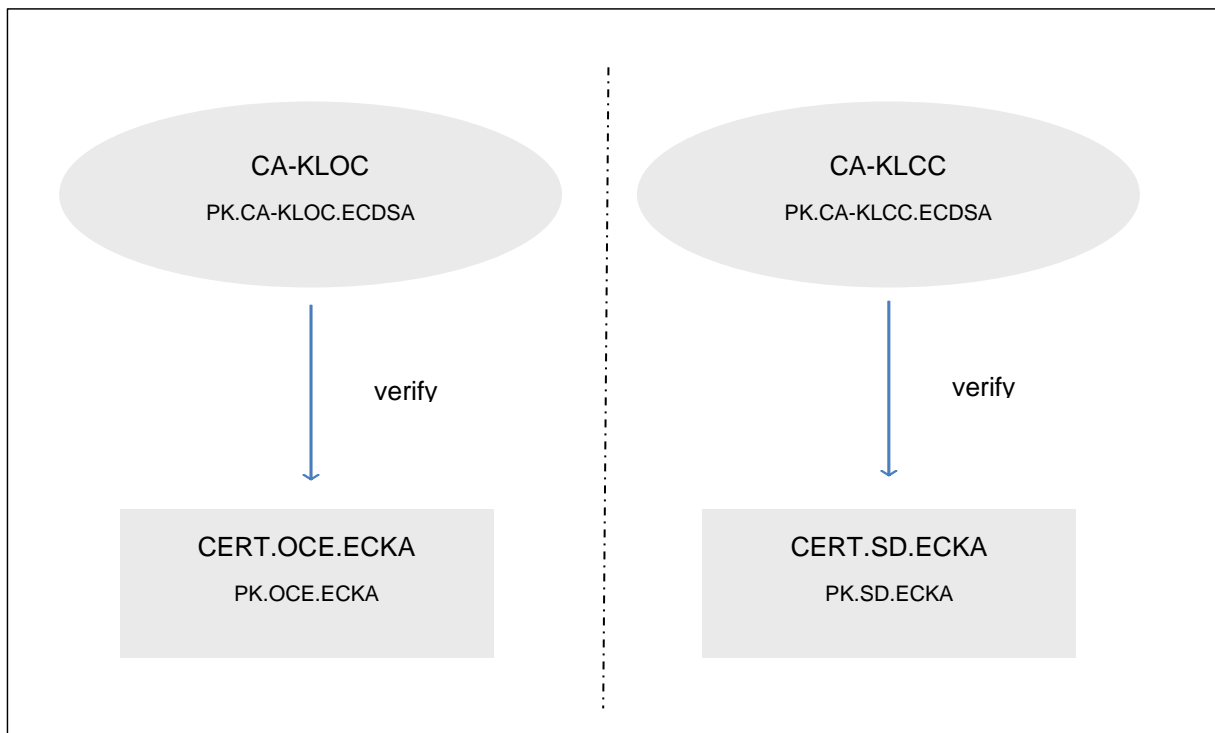
The Security Domain and the OCE may need to traverse and verify a chain of certificates from established trust points down to each other's public key.

By default, the Security Domain expects the first certificate presented for verification to be signed by the CA-KLOC, and each subsequent certificate to be signed by the public key validated with the previous certificate.

By default, the OCE expects the first certificate presented for verification to be signed by the CA-KLCC, and each subsequent certificate to be signed by the public key validated with the previous certificate.

The example shown in Figure 3-1 is a simple case where the CA-KLOC directly certifies the public key of the Off-Card Entity and the CA-KLCC directly certifies the public key of the Security Domain.

Figure 3-1: Single Certificate (No Chaining)

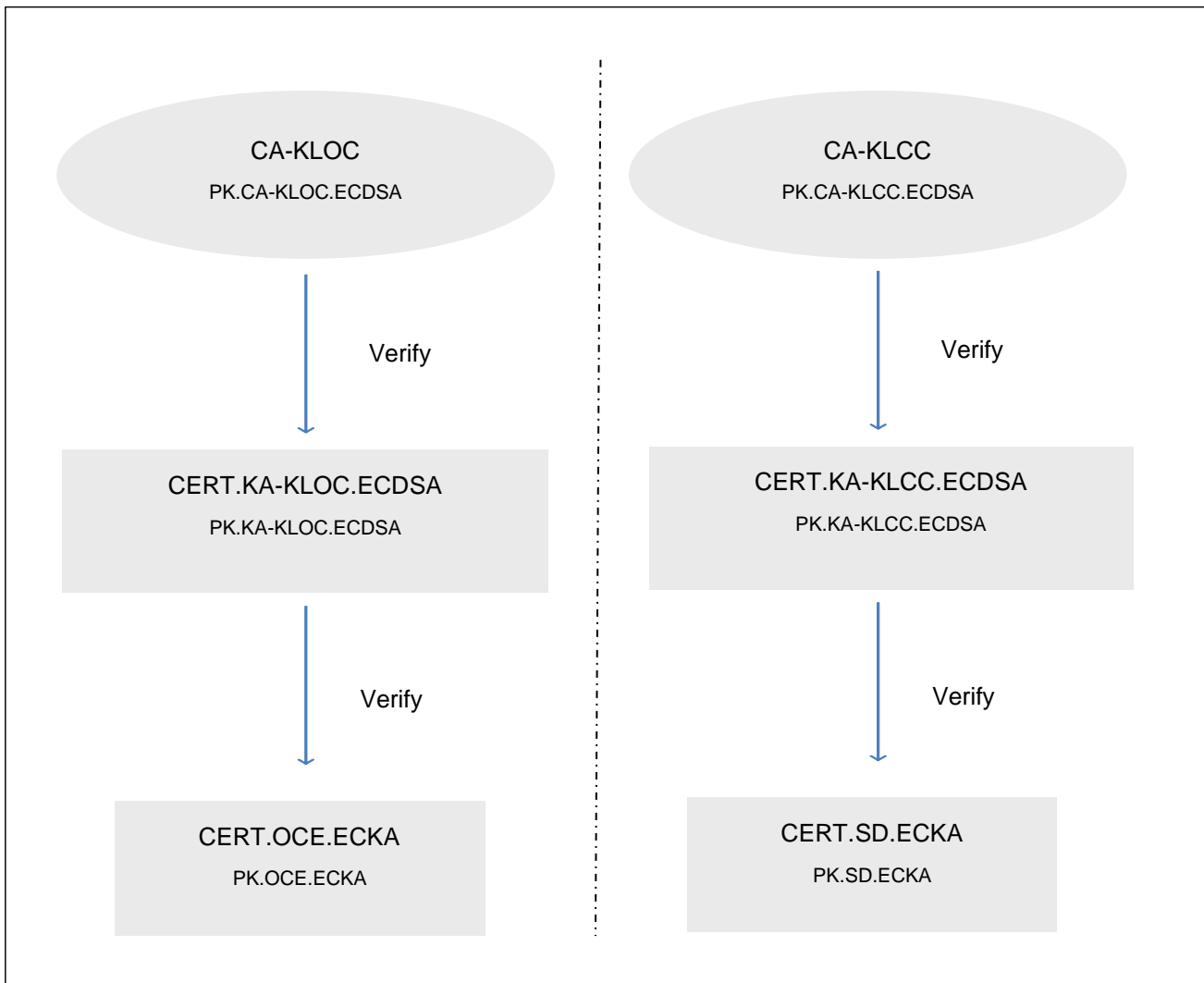


The example shown in Figure 3-2 illustrates the usage of a certificate chain.

The CA-KLOC certifies the KA certificate (CERT.KA-KLOC.ECDSA) which becomes a subordinate CA. The KA then certifies the last certificate (CERT.OCE.ECKA) which contains the public key of the OCE used for key agreement.

In the same way, the CA-KLCC certifies the KA certificate (CERT.KA-KLCC.ECDSA) which becomes a subordinate CA. The KA then certifies the last certificate (CERT.SD.ECKA) which contains the public key of the SD used for key agreement.

Figure 3-2: Certificate Chain



4 Secure Channel Protocol Usage

4.1 Protocol Overview

Before setting up a secure channel, the OCE has to retrieve 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. In particular, in the case of SCP11c the OCE can retrieve the SD certificate(s) from the issuer(s) in advance through another channel out of scope of this document.

Figure 4-1: Initial Certificate Retrieval

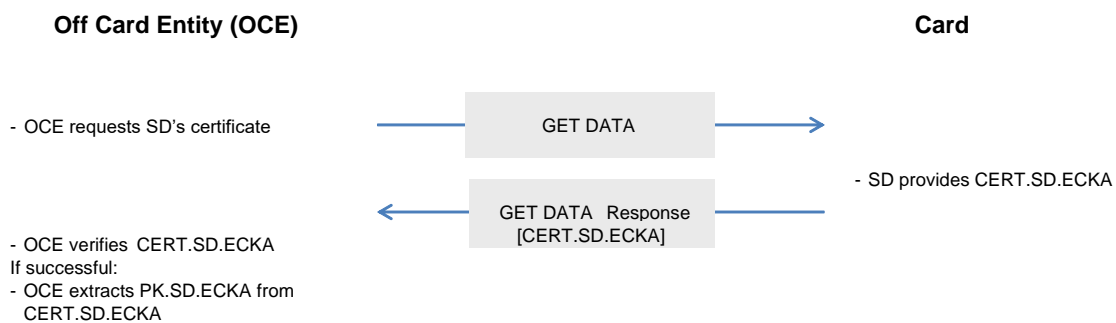


Figure 4-2, Figure 4-3, and Figure 4-4 provide an overview of the three variants of SCP11.

SCP11a and SCP11c provide 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. For SCP11a, depending 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 authenticates the SD to the OCE but not vice versa, the PERFORM SECURITY OPERATION is not required.

SCP11a uses ephemeral keys for providing forward secrecy, while SCP11c does not use an ephemeral key on the SD side. Therefore, SCP11c can be used in a static command script, but it does not provide forward secrecy for the SD.

Figure 4-2: SCP11a Protocol Overview

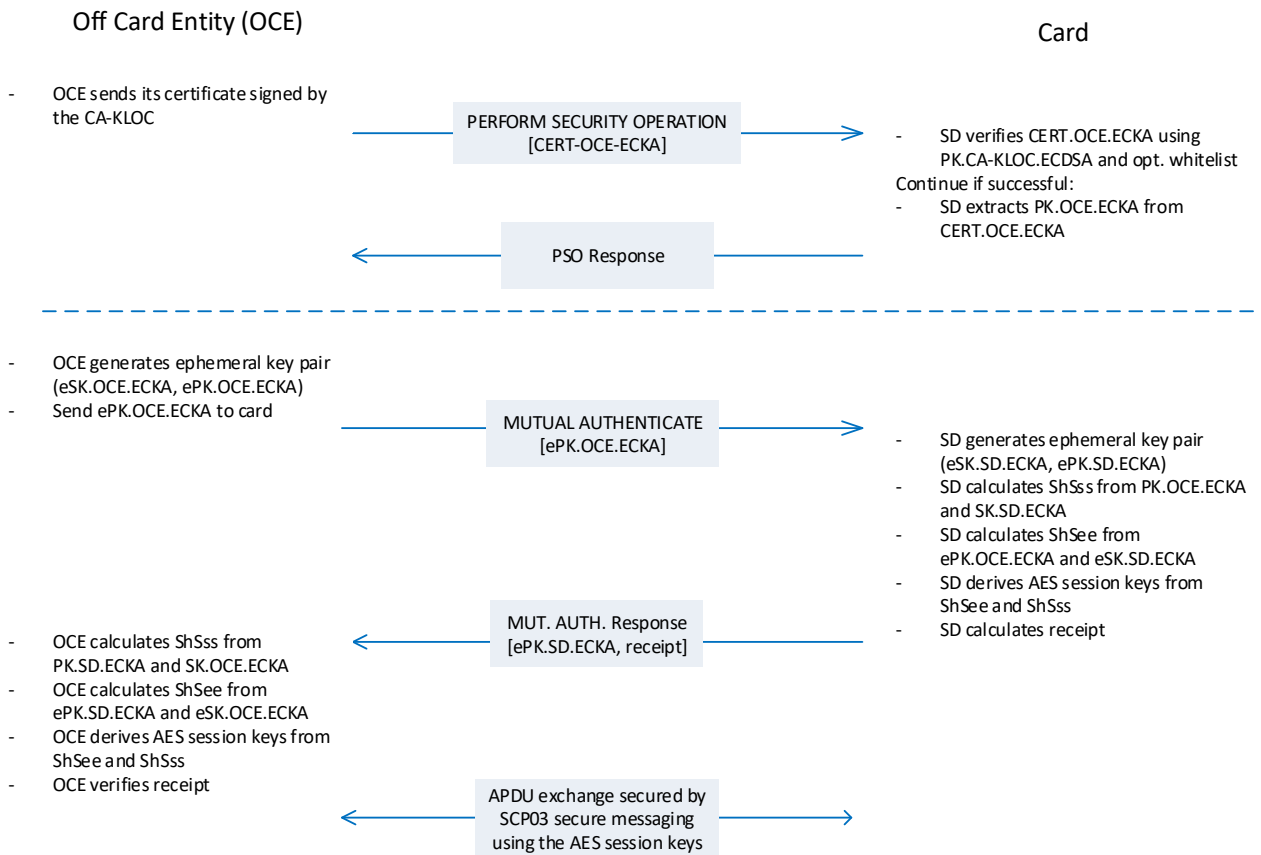


Figure 4-3: SCP11b Protocol Overview

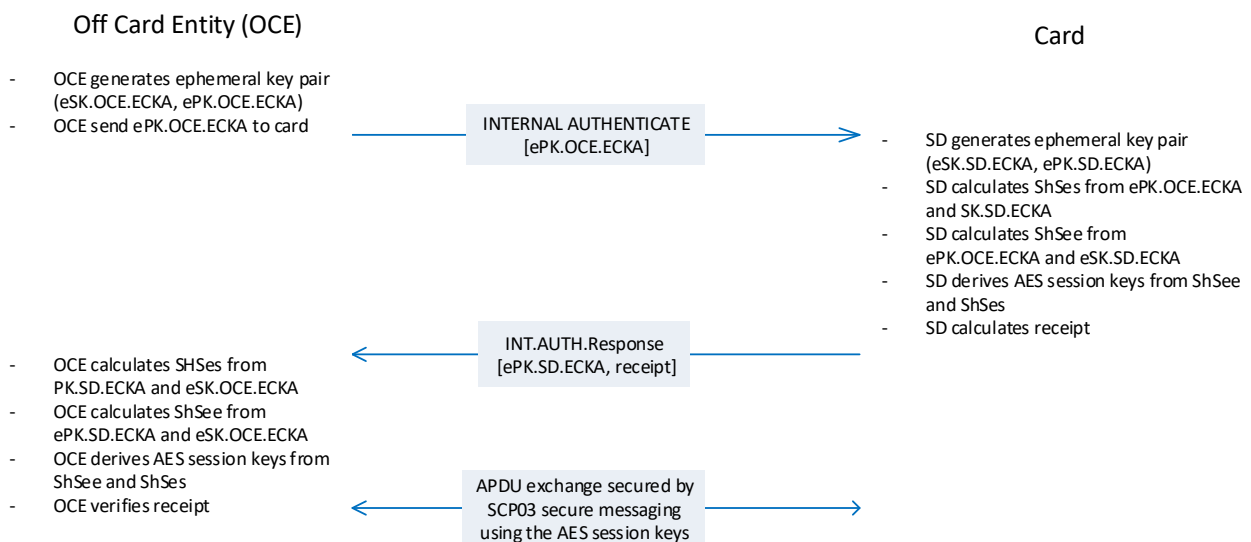
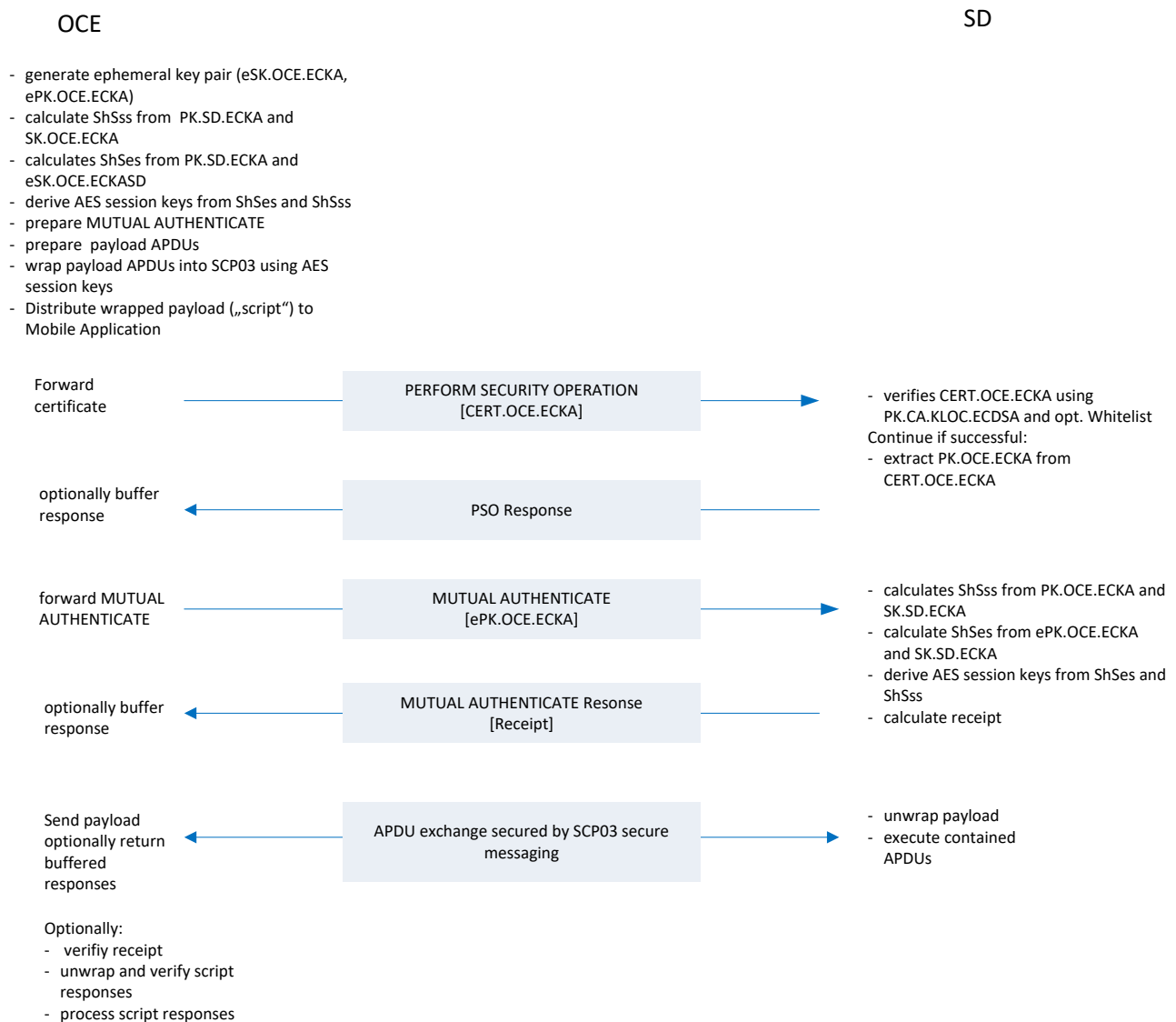


Figure 4-4: SCP11c Protocol Overview



The OCE may be a remote entity or may be split into a remote entity that precomputes data and a local agent (e.g. a mobile application on a smartphone) that forwards data to the SD. The OCE may precompute the commands or receive precomputed commands from another entity; this is out of scope of this document. After the execution of the APDU commands has finished, the local agent may return response data to the remote entity if required.

Note: If an OCE wishes to support several different Card Group IDs, e.g. because it wishes to support different issuers, it has to precompute several scripts using different CERT.SD.ECKA. In this case, the OCE may have to obtain several CERT.SD.ECKA from different vendors in order to precompute these scripts. The distribution infrastructure of these certificates is out of scope of this document.

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, SCP11b, and SCP11c are given in section 4.3 and section 4.4.

In SCP11, 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	1: SCP11a supported
						X		1: SCP11b supported
					X		1	1: SD persistently stores PK.OCE.ECKA (only applicable to SCP11a)
				X				1: Certificate chain supported
			X					1: SCP11c supported
		X						1: SCP11c authorization mechanism supported ('BF20')
	X							0: Secure messaging using S8 mode (see [Amd D]) 1: Secure messaging using S16 mode (see [Amd D])
X								Reserved; see [GPCS] section E.1.1

An implementation may support one or two or all variants of SCP11.

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 and SCP11c only: The OCE authenticates itself 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. However, if the whitelist is used in this way, special care shall be taken never to empty/remove the whitelist (i.e. if created, the whitelist shall always contain at least one certificate) because no restrictions apply (i.e. all certificates are accepted) once a whitelist is removed.

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 and SCP11c provide Mutual Authentication between the OCE and the SD.

SCP11b provides Authentication of the SD to the OCE only.

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 or SCP11c is currently in use (see section 4.7).

Note: When executing the MUTUAL AUTHENTICATE command, the SD will be able to distinguish between SCP11a and SCP11c by evaluating tag '90' in the command data.

4.4 Message Integrity and Data Confidentiality

Message Integrity and Data Confidentiality is achieved by 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

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.

SCP11a and SCP11b 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.

SCP11c does not provide Forward Secrecy because only one ephemeral key pair is used (generated on the OCE side).

4.6 Session Replay

For SCP11a and SCP11b, session replay is not possible as new session keys are generated for each session and the session key generation process includes an ephemeral key pair generated by the target SD.

For SCP11c, session replay is possible as the randomness of session keys only depends on an ephemeral key pair generated by the OCE. Therefore, operations performed within an SCP11c session are controlled as described below:

- The usage of PUT KEY, DELETE [key(s)], and SET STATUS commands shall not be allowed. The STORE DATA command is allowed but should not be used to load keys out of a secure and controlled environment. For use cases where keys need to be set up, it is recommended to use scenario #1 or scenario #3 from [Amd A].
- If the Security Level is ANY_AUTHENTICATED, the usage of some APDU commands and TLVs/DGIs in the STORE DATA command is only allowed if explicitly authorized by CERT.OCE.ECKA. See Annex B and Table B-3 for details.

Only one SCP11c session per SD is allowed at a given time.

4.7 API and Security Level

An SD supporting SCP11 shall implement the `SecureChannel` interface of the API specified in [GPCS]. An application associated with 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;
 - After successful processing of a MUTUAL AUTHENTICATE command (for SCP11a and SCP11c):
 - Either AUTHENTICATED or ANY_AUTHENTICATED; and
 - C_MAC, R_MAC, C_DECRYPTION, and/or R_ENCRYPTION depending on the key usage qualifier specified in the MUTUAL AUTHENTICATE command (see Table 6-17).
 - After successful processing of an INTERNAL AUTHENTICATE command (for SCP11b), C_MAC, R_MAC, C_DECRYPTION, and/or R_ENCRYPTION depending on the key usage qualifier specified in the INTERNAL AUTHENTICATE command (see Table 6-23).

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.*

Note: *For SCP11c, card content management operations which are listed in the certificate are allowed in addition to those granted by the Security Level. For details refer to Annex B.*

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.

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.*

4.8 Protocol Rules

In accordance with the general rules described in [GPCS] section 10, 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 any of the following:
 - 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	SCP11a and SCP11b only Mandatory
ePK.SD.ECKA	Ephemeral public key of the SD used for key agreement	see below	SCP11a and SCP11b only 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 (or subordinate KA-KLCC)	see below	Mandatory
SK.OCE.ECKA	Private key of the OCE used for key agreement	see below	SCP11a and SCP11c only
PK.OCE.ECKA	Public key of the OCE used for key agreement	see below	SCP11a and SCP11c only
CERT.OCE.ECKA	Certificate containing the public key of the OCE used for key agreement, signed by the CA-KLOC (or subordinate KA-KLOC)	see below	SCP11a and SCP11c only
SK.CA-KLOC.ECDSA	Private key of the CA-KLOC used for signing certificates	see below	SCP11a and SCP11c only
PK.CA-KLOC.ECDSA	Public key of the CA-KLOC used for verifying certificates	see below	SCP11a and SCP11c only
CERT.KA-KLOC.ECDSA	Certificate containing the public key of the KA-KLOC used for verifying certificates, signed by the CA-KLOC (or another intermediate KA-KLOC)	see below	SCP11a and SCP11c only Optional
SK.KA-KLOC.ECDSA	Private key of the KA-KLOC used for signing certificates	see below	SCP11a and SCP11c only Optional
PK.KA-KLOC.ECDSA	Public key of the KA-KLOC used for verifying certificates	see below	SCP11a and SCP11c only Optional

Key	Usage	Length	Remark
CERT.KA-KLCC.ECDSA	Certificate containing the public key of the KA-KLCC used for verifying certificates, signed by the CA-KLCC (or another intermediate KA-KLCC)	see below	SCP11a and SCP11c only Optional
SK.KA-KLCC.ECDSA	Private key of the KA-KLCC used for signing certificates	see below	SCP11a and SCP11c only Optional
PK.KA-KLCC.ECDSA	Public key of the KA-KLCC used for verifying certificates	see below	SCP11a and SCP11c only Optional

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 described in [GPCS] section B.4.1.

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 only
- KID '13' for SK.SD.ECKA used for SCP11b
- KID '14' for the optional static Key-DEK used with SCP11b only
- KID '15' for SK.SD.ECKA used for SCP11c
- KID '16' for the optional static Key-DEK used with SCP11c only
- KID from '20' to '2F' for additional PK.CA-KLOC.ECDSA

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 6.7). Each SK.SD.ECKA must have one associated CERT.SD.ECKA.

If certificate chaining and one or more Key Authorities are used, the public key extracted from each intermediary certificate should not be stored persistently but shall be used to verify the next certificate in the chain down to PK.OCE or PK.SD.

Within the same SD, different PK.CA-KLOC.ECDSA may be stored and used to verify CERT.OCE.ECKA, or the first CERT.KA-KLOC.ECDSA of a certificate chain if certificate chaining is used. A range of Key Identifiers is defined for storing different PK.CA-KLOC.ECDSA.

When contained in a command or a response, static or ephemeral public keys shall be formatted using uncompressed encoding as specified in [TR 03111] section 3.1.1, 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 ShSee, ShSss, ShSes, ShSse, 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 no static Key-DEK is 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 a 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.

For SCP11a and SCP11b, 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

For SCP11c, four 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 AUTHENTICATE command:

- C-MAC and R-MAC only
- C-DECRYPTION, R-ENCRYPTION, C-MAC, and R-MAC
- C-DECRYPTION and C-MAC
- C-DECRYPTION, 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 with 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 with 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/c is used.

Table 6-1: SCP11 Command Support

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

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
–	–	–	–	–	–	XX	00: Indicates SCP11b 01: Indicates SCP11a 11: Indicates SCP11c 10: RFU
–	–	–	–	–	X	–	0: Do not include Host and Card ID / Card Group ID in key derivation process 1: Include Host and Card ID / Card Group ID in key derivation process
X	X	X	X	X	–	–	RFU (0)

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

Note: SCP11a and SCP11b use Card ID, while SCP11c uses Card Group ID.

6.2 GET DATA (ECKA Certificate) Command

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

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(s)) 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 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(s)) Response

Tag	Length	Value Description		MOC	
'BF 21'	Var	SCP11 certificate store		M	
		Tag	Length	Value Description	MOC
		'7F 21'	Var	CERT.KA-KLCC.ECDSA	O
		'7F 21'	Var	O
		'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 A]. Tag '42' identifies the owner of the SD; tag '45' identifies the Security Domain Image Number.

For each certificate, the OCE shall verify at least the following:

- The signature of the certificate, using:
 - PK.CA-KLCC.ECDSA for the 1st (or only) certificate
 - The public key extracted from the previous CERT.KA-KLCC.ECDSA, if certificate chaining is used
- The Expiration Date, to ensure that the certificate is still valid
- The correctness of the Key Usage (i.e. shall be '00 80' (ECKA) for the last (or only) certificate)

In addition, the OCE should check either a whitelist or a revocation list for the SD's certificates. Such a whitelist or a revocation list 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.

6.2.1 Response Message

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

If the response exceeds 256 bytes, it shall be chained as described in [GPCS] section 11.1.5.2, using the '61xx' status word, and multiple GET RESPONSE commands should be used to retrieve the remaining response data.

6.3 GET DATA (CA-KLOC KID-KVN) Command

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

It is used by the OCE to retrieve from the SD the KID and the KVN corresponding to a given CA-KLOC Identifier.

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 '00 83'.

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

Table 6-5: Data Field of GET DATA (CA-KLOC KID-KVN) Command

Tag	Length	Value Description		MOC	
'A6'	Var	Control Reference Template		M	
		Tag	Length	Value Description	MOC
		'42'	1-16	CA-KLOC Identifier	M

The SD shall return the KID and KVN linked to the public key PK.CA-KLOC.ECDSA referenced by the CRT, encapsulated according to the following table.

Table 6-6: Data Field of GET DATA (CA-KLOC KID-KVN) Response

Tag	Length	Value Description	MOC
'83'	2	byte 1: Key Identifier byte 2: Key Version Number	M

6.4 PERFORM SECURITY OPERATION Command

6.4.1 Definition and Scope

The PERFORM SECURITY OPERATION command is used to submit CERT.OCE.ECKA, or possibly a chain of certificates ending with CERT.OCE.ECKA. This is required as a precondition to the initiation of a SCP11a or SCP11c secure channel.

This command does not terminate an ongoing secure channel session.

For each certificate:

- The certificate's signature shall be verified.
- The Certificate Serial Number (CSN) shall be checked as further explained below.
- The Key Usage value shall be checked.
- The structure of the public key, including the existence of the referenced key parameters, shall be checked and the public key shall be extracted.
- All other fields of the certificate may be ignored.

If certificate chaining is not used, only CERT.OCE.ECKA is submitted and:

- CERT.OCE.ECKA shall be verified using the referenced PK.CA-KLOC.ECDSA (see P1 and P2 parameters).
- If no whitelist is defined for the referenced PK.CA-KLOC.ECDSA (see section 6.8), the CSN of CERT.OCE.ECKA shall not be checked. Otherwise, if this CSN is not referenced in the whitelist, then the certificate shall be rejected.

When certificate chaining and intermediate Key Authority(s) are used:

- The payload is composed of a chain of certificates starting from the first (or only) CERT.KA-KLOC.ECDSA and ending with CERT.OCE.ECKA, and this command shall be used to submit one certificate at a time (see usage of P1 and P2 parameters).
- The first (or only) CERT.KA-KLOC.ECDSA shall be verified using the referenced PK.CA-KLOC.ECDSA. Each subsequent certificate in the chain shall be verified using the public key extracted from the certificate immediately preceding it. In particular, the reference to PK.CA-KLOC.ECDSA given by the P1 and P2 parameters is only relevant for the first (or only) CERT.KA-KLOC.ECDSA, and shall be ignored for subsequent certificates.
- If no whitelist is defined for the referenced PK.CA-KLOC.ECDSA (see section 6.8), CSNs found in the certificate chain shall not be checked. Otherwise, if the CSN of the first (or only) CERT.KA-KLOC.ECDSA is not referenced in the whitelist, then the certificate (and the entire certificate chain) shall be rejected. Other CSNs found in the certificate chain shall not be checked (i.e. other certificates following in the chain are not required to be referenced in this whitelist in order to be accepted).
- The whitelist only contains CSNs of intermediate Key Authorities directly signed by the CA. Thus all certificates issued by any whitelisted intermediate Key Authority will be accepted.

If these verifications are successful, the SD shall eventually extract both PK.OCE.ECKA and the Subject Identifier (tag '5F20') from CERT.OCE.ECKA and, depending on the implementation option (see section 4.2), store them persistently (only if tag 'BF20' was not found in the certificate) or temporarily for use in any subsequent MUTUAL AUTHENTICATE command. 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 1: The Subject ID shall subsequently be used to determine whether the established security level is AUTHENTICATED or ANY_AUTHENTICATED.

Note 2: 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.

6.4.2 Command Message

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

Table 6-7: 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 and command chaining information
P2	'xx'	Key Identifier and certificate chaining information
Lc	'xx'	Length of data field
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.

6.4.2.1 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 to 7. Bit 8 is used for the command chaining information.

The command chaining information indicates whether the data field of the command is an intermediate or the last block of the currently submitted certificate.

Table 6-8: Values of Reference Control Parameter P1

b8	b7	b6	b5	b4	b3	b2	b1	Description
X								Command chaining information: 1: More command(s) 0: Last command
	X	X	X	X	X	X	X	Key version number of the PK.CA-KLOC.ECDSA

If the certificate doesn't exceed 255 bytes, the command chaining information (b8) shall be set to 0 (Last command). Otherwise, bit 8 shall be set to 1 (More command(s)) until the last block of the certificate.

6.4.2.2 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 to 7. Bit 8 is used for the certificate chaining information.

The certificate chaining information indicates whether the submitted certificate (in the data field of the command) is an intermediate or the last certificate of the chain i.e. intermediate CERT.KA-KLOC.ECDSA or final CERT.OCE.ECKA.

Table 6-9: Values of Reference Control Parameter P2

b8	b7	b6	b5	b4	b3	b2	b1	Description
X								Certificate chaining information: 1: More certificate(s) 0: Last certificate
	X	X	X	X	X	X	X	Key identifier of the PK.CA-KLOC.ECDSA

Bit 8 shall be set to 0 (Last certificate) if the submitted certificate is CERT.OCE.ECKA, i.e. the last certificate. Otherwise, for intermediate certificates, bit 8 shall be set to 1 (More certificate(s) expected).

If bit 8 is set to 1 and SCP options (see Table 4-1) indicate that certificate chains are not supported, then a response of '6A86' shall be returned.

6.4.2.3 Example of Using P1P2

Given a certificate chain (CERT1...CERTn and CERT.OCE.ECKA) and certificates exceeding 255 bytes, the scenario is as follows:

Table 6-10: Example of Using P1P2

	Cmd N°	P1.b8	P2.b8
CERT1	0	More command(s)	More certificate(s)
	...		
	N-1		
	N	Last command	
...			
CERTn	0	More command(s)	More certificate(s)
	...		
	N-1		
	N	Last command	
CERT.OCE.ECKA	0	More command(s)	Last certificate
	...		
	N-1		
	N	Last command	

6.4.2.4 Data Field Sent in the Command Message

The data field of the command message shall contain part or all of the data described in the following table. Such data may be split across several consecutive PERFORM SECURITY OPERATION commands as described by the P1 and P2 parameters.

Table 6-11: PERFORM SECURITY OPERATION Command Data

Tag	Length	Value Description	MOC
'7F21'	Var	Certificate (CERT.KA-KLOC.ECDSA)	O
...			O
'7F21'	Var	Certificate (CERT.OCE.ECKA)	M

Where certificate(s) have the following format:

Table 6-12: Certificate Format

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 (or KA-KLOC) Identifier	M
		'5F20'	1-16	Subject Identifier	M
		'95'	1-2	Key Usage: <ul style="list-style-type: none"> '82': Digital signature verification (CERT.KA-KLOC.ECDSA) '00 80': Key agreement (CERT.OCE.ECKA) 	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
		'BF20'	Var	Authorizations under SCP11c	O
		'7F49'	Var	Public Key – for details, see tables below	M
'5F37'	Var	Signature	M		

Tags '53' and '73' are mutually exclusive.

When the Security Level established for an SCP11c secure channel session is ANY_AUTHENTICATED, tag 'BF20' allows to selectively authorize the actions that may be performed during the session (i.e. superseding the rules defined in [GPCS] Table 11-2). For details, see Annex B. This tag may be present for a key agreement certificate (i.e. value '00 80' for tag '95') which is to be used to open an SCP11c session. It shall be absent in a key agreement certificate which is to be used to open an SCP11a session. Finally, it shall be absent in a signature verification certificate (i.e. value '82' for tag '95').

If a Security Domain does not support the 'BF20' authorization mechanism (as indicated by its SCP11 "i" option) and tag 'BF20' is present, then the command shall fail with an error of '6A81'. In this case, card content management commands may only be processed if the Security Level AUTHENTICATED is established.

The TLV-encoded data listed in the following table are signed off-card with SK.CA-KLOC.ECDSA for the CERT.KA-KLOC.ECDSA (or SK.KA-KLOC.ECDSA for the next certificate), and the last certificate (CERT.OCE.ECKA) is signed with the private key associated with the previous certificate. The signature is stored in the content of tag '5F37' (signature), as described in [Amd A].

Table 6-13: Data Signed to Generate the Certificate

Tag	Length	Value Description	MOC
'93'	1-16	Certificate Serial Number	M
'42'	1-16	CA-KLOC (or KA-KLOC) Identifier	M
'5F20'	1-16	Subject Identifier	M
'95'	1-2	Key Usage: <ul style="list-style-type: none"> '82': Digital signature verification (CERT.KA-KLOC.ECDSA) '00 80': Key agreement (CERT.OCE.ECKA) 	M
'5F25'	4	Effective Date (YYYYMMDD, BCD format)	C
'5F24'	4	Expiration Date (YYYYMMDD, BCD format)	M
'53' or '73'	1-127	Discretionary Data	C
'BF20'	Var	Authorizations under SCP11c	C
'7F49'	Var	Public Key	M

Tags '5F25', '53', '73', and 'BF20' shall be included in signed data if they are present in the OCE certificate.

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

Table 6-14: 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

Note: In our example, CERT1 is verified using the preloaded PK.CA-KLOC.ECDSA and CERT.OCE.ECKA is verified using the extracted public key (PK.CERTn) from the CERTn.

6.4.3 Response Message

6.4.3.1 Data Field Returned in the Response Message

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

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: 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'	'81'	'BF20' authorization mechanism not supported
'6A'	'88'	Referenced PK.CA-KLOC.ECDSA not found
'6A'	'86'	Certificate chains not supported

6.5 MUTUAL AUTHENTICATE Command

6.5.1 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 any ongoing secure channel session (whichever secure channel protocol is currently used) on the same logical channel and if the command is successful, initiates a new secure channel session.

If no PK.OCE.ECKA was provided earlier to the SD using a PERFORM SECURITY OPERATION command or if a whitelist was updated since PK.OCE.ECKA was provided (see section 6.8), 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 should check that the SD used the correct PK.OCE.ECKA by verifying the receipt generated by the SD.

6.5.2 Command Message

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

Table 6-16: 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 of data field
Data	'xx xx...'	Data for key establishment
Le	'00'	

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

6.5.2.1 Reference Control Parameter P1

Reference control parameter P1 references the Key Version Number of the SK.SD.ECKA. It is coded on bits 1 to 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 to 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: 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 <ul style="list-style-type: none"> '34' (secure messaging with C-MAC and R-MAC) '3C' (secure messaging with C-MAC, C-ENC, R-MAC, and R-ENC) '1C' (secure messaging with C-MAC and C-ENC, only for SCP11c) '74' (secure messaging with C-MAC, C-ENC, and R-MAC, only for SCP11c) (See [GPCS] Table 11-17)	M
		'80'	1	Key Type according to [GPCS] Table 11-16 <ul style="list-style-type: none"> '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.

The key usage qualifier (tag '95') shall be used to specify the Security Level of the SCP11 session. The usage of values '1C' and '74' is only allowed for SCP11c.

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.

In the case of SCP11a:

- If the SD also supports SCP11c, it shall check that the CERT.OCE.ECKA provided by the previous PERFORM SECURITY OPERATION command did not contain tag 'BF20'.
- 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 ShSss according to section 3.1.1.
- The SD shall use ePK.OCE.ECKA and eSK.SD.ECKA to generate the shared secret ShSee according to section 3.1.1.

- The SD shall concatenate ShSee and ShSss to form the shared secret ShS which constitutes the input for the Key Derivation process.

In the case of SCP11c:

- The SD shall remove all associated objects that have the “pending completion” mark.
- The SD shall use PK.OCE.ECKA and SK.SD.ECKA to generate the shared secret ShSss according to section 3.1.1.
- The SD shall use ePK.OCE.ECKA and SK.SD.ECKA to generate the shared secret ShSes according to section 3.1.1.
- The SD shall concatenate ShSes and ShSss 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:
 - In the case of SCP11a and SCP11b: HostID-LV, SIN-LV, and SDIN-LV
 - In the case of SCP11c: HostID-LV and Card Group ID-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]).

Card Group ID-LV is the length and the value field of tag '5F20' (subject identifier) in CERT.SD.ECKA.

Note: *The OCE may need to obtain CERT.SD.ECKA before creating the SCP11c script to obtain PK.SD.ECKA.*

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 6-18: KeyData Assignment

<i>KeyData</i>	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 6-19. The receipt key shall be deleted after calculating the receipt.

Table 6-19: Input Data for Receipt Calculation

Tag	Length	Data Element	MOC
'A6'	Variable	CRT TLV with all sub TLVs as provided in the MUTUAL AUTHENTICATE command	M
'5F49'	Variable	ePK.OCE.ECKA	M
'5F49'	Variable	SCP11a: ePK.SD.ECKA SCP11c: PK.SD.ECKA	M

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 data objects listed in the following table.

Table 6-20: MUTUAL AUTHENTICATE Response Data

Tag	Length	Value Description	MOC
'5F49'	Variable	SCP11a: ePK.SD.ECKA SCP11c: PK.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 6-21: 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.ECKA / PK.OCE.ECKA / SIN / SDIN
'69'	'85'	Attempt to initiate an SCP11c session but one SCP11c session is already ongoing on another logical channel and the implementation does not support multiple SCP11c sessions

6.6 INTERNAL AUTHENTICATE Command

6.6.1 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.

6.6.2 Command Message

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

Table 6-22: 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 of data field
Data	'xx xx...'	Data for key establishment
Le	'00'	

6.6.2.1 Reference Control Parameter P1

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

6.6.2.2 Reference Control Parameter P2

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

6.6.2.3 Data Field Sent in the Command Message

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

Table 6-23: 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 <ul style="list-style-type: none"> '34' (secure messaging with C-MAC and R-MAC only) or '3C' (secure messaging with C-MAC, R_MAC, and ENCRYPTION) (See [GPCS] Table 11-17)	M
		'80'	1	Key Type according to [GPCS] Table 11-16 <ul style="list-style-type: none"> '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 6.5.2.3, with the following modifications:

- The SD shall generate an ephemeral key pair eSK.SD.ECKA and ePK.SD.ECKA.
- The SD shall use ePK.OCE.ECKA and SK.SD.ECKA to generate the shared secret ShSes according to section 3.1.1.
- The SD shall use ePK.OCE.ECKA and eSK.SD.ECKA to generate the shared secret ShSee according to section 3.1.1.
- The SD shall concatenate ShSee and ShSes to form the shared secret ShS which constitutes the input for the Key Derivation process.

6.6.3 Response Message

6.6.3.1 Data Field Returned in the Response Message

The data field of the response message shall contain the data objects listed in the following table.

Table 6-24: INTERNAL AUTHENTICATE Response Data

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

6.6.3.2 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 6-25: INTERNAL AUTHENTICATE Error Conditions

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

6.7 STORE DATA (ECKA Certificate) Command

The STORE DATA command is defined in [GPCS] section 11.11. 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 6-26: Data Field of STORE 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 byte 2: Key Version Number	M
'BF21'	Var	SCP11 certificate store (see section 6.2, Table 6-4)			M

If certification chaining and one or more Key Authorities are used, the SCP11 certificate store shall contain all the certificates in suitable verification order, i.e. starting from the one signed by CA-KLCC and ending with CERT.SD.ECKA. Except for the first certificate (that shall be verified using PK.CA-KLCC.ECDSA), each certificate shall be verified by the public key extracted from the certificate preceding it.

After successful execution of the command, the SCP11 certificate store is linked to the private key SK.SD.ECKA referenced by the CRT. This certificate store may contain a chain of certificates ending with CERT.SD.ECKA as shown in section 6.2, Table 6-4.

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'.

6.8 STORE DATA (Whitelist) Command

The STORE DATA command is defined in [GPCS] section 11.11. 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 three BER-TLVs as defined in the following table.

Update of the whitelist shall only be accepted if the Security Level is AUTHENTICATED.

Note: Security Level AUTHENTICATED cannot be achieved with SCP11b.

Table 6-27: 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 byte 2: Key Version Number	M
'92'	2	Whitelist counter (2-byte positive integer up to 32767)		C	
'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. On the other hand, the update of a PK.CA-KLOC.ECDSA shall have no impact on a linked whitelist.

If the SD supports SCP11c (according to its “i” parameter), then:

- The SD shall maintain an internal anti-replay whitelist counter for each PK.CA-KLOC.ECDSA. This counter shall never be reset, even if the whitelist is deleted (i.e. the same counter shall be used for any whitelist subsequently added for the same PK.CA-KLOC.ECDSA).
- Tag '92' shall be present in the command if it is received in an SCP11c session and shall not be present otherwise.

- If this command is received in an SCP11c session, then:
 - If any of the following conditions is true:
 - The whitelist counter (tag '92') is missing in the command, or
 - The value of the internal counter is higher than or equal to the whitelist counter sent in the command, or
 - The internal counter value has reached its maximum value (32767),
...then an error of '6A80' shall be returned.
 - If the update of the whitelist is successful, then the value of the internal counter shall be updated to the value of tag '92'. Notice that the whitelist and the internal counter shall be updated atomically in the same transaction.

6.9 STORE DATA (CA-KLOC Identifier) Command

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

To store or replace a CA-KLOC Identifier linked to a public key, the data field of the command message shall contain a BER-TLV as defined in the following table.

Table 6-28: Data Field of STORE DATA (CA-KLOC Identifier) Command

Tag	Length	Value Description		MOC	
'A6'	Var	Control Reference Template		M	
		Tag	Length	Value Description	MOC
		'42'	1-16	CA-KLOC Identifier	M
'83'	2	byte 1: Key Identifier byte 2: Key Version Number		M	

After successful execution of the command, the CA-KLOC Identifier store is linked to the public key PK.CA-KLOC.ECDSA referenced by the CRT.

When PK.CA-KLOC.ECDSA is deleted or replaced, the CA-KLOC Identifier store shall be automatically deleted by the SD.

If the referenced PK.CA-KLOC.ECDSA does not exist in the SD, the command shall be rejected with error condition '6A88'.

Table 6-29: STORE DATA Error Conditions

SW1	SW2	Meaning
'6A'	'80'	Incorrect values in command data
'6A'	'88'	The referenced PK.CA-KLOC.ECDSA does not exist in the SD

Annex A OCE Authentication for SCP11b

SCP11b provides authentication of the card to the 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 TEE Trusted User Interface API, [TEE TUI]).

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

Note: Assuming PIN verification would be implemented by the SD, whether successful verification should have an impact on the actual security level of the SCP11b session (i.e. AUTHENTICATED or ANY_AUTHENTICATED) is out of scope and policy dependent.

This annex provides an example of 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: Normal operation
P2	'00'	Reference control parameter P2: 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

Annex B Usage of SCP11c

This document introduces some changes to the behavior of the card if SCP11c has been set up successfully. These changes are described in this annex.

B.1 Authentication Level

When SCP11c has been set up successfully, the authentication level of the Security Domain shall be set accordingly for the processing of the commands described in this annex.

The API method `SecureChannel.getSecurityLevel()` shall return a value from Table B-1, depending on:

- Key Usage Qualifier
- Common name (tag '5F20') in the OCE certificate

Table B-1: Authentication Levels in SCP11c

Common Name	Key Usage Qualifier	Authentication Level
Does not match owner of the Security Domain	'34'	ANY_AUTHENTICATED C_MAC R_MAC
	'3C'	ANY_AUTHENTICATED C_MAC C_DECRYPTION R_MAC R_ENCRYPTION
	'1C'	ANY_AUTHENTICATED C_MAC C_DECRYPTION
	'74'	ANY_AUTHENTICATED C_MAC C_DECRYPTION R_MAC
Matches owner of the Security Domain	'34'	AUTHENTICATED C_MAC R_MAC
	'3C'	AUTHENTICATED C_MAC C_DECRYPTION R_MAC R_ENCRYPTION
	'1C'	AUTHENTICATED C_MAC C_DECRYPTION
	'74'	AUTHENTICATED C_MAC C_DECRYPTION R_MAC

If the Security Level is AUTHENTICATED, commands shall be accepted as described in [GPCS] Table 11-2, except commands that are forbidden as described in section 4.6.

If the Security Level is ANY_AUTHENTICATED, commands shall be accepted depending on the authorizations defined in tag 'BF20' (if present) in the certificate CERT.OCE.ECKA.

B.2 Limitations on Commands Received through SCP11c

SCP11c is meant to be used in scenarios where the same static script is to be applied to many cards.

When an SCP11c secure channel session is established with the AUTHENTICATED Security Level, Security Domain APDU commands are accepted as described in [GPCS] Table 11-2, except commands which are always disallowed as described in section 4.6 of this document.

When an SCP11c secure channel session only establishes the ANY_AUTHENTICATED Security Level, the usage of Security Domain APDU commands is only allowed if authorized by tag 'BF20' in CERT.OCE.ECKA.

Note: The successful processing of the APDU commands present in an SCP11c script may also depend on the privileges of the Security Domain processing the script (e.g. Authorized Management, Delegated Management, etc.).

In order to specify these authorizations, the following tags may be used in the OCE certificate:

Table B-2: Contents of Tag 'BF20' in OCE Certificate

Tag	Length	Value Description	MOC		
'E3'	Var	GlobalPlatform Registry related data			
		Tag	Length	Value Description	MOC
		'8A'	5-16	<p>AID prefix bytes used to control which Application may be installed, and/or which ELF may be loaded.</p> <p>Only ELF and Application AID values matching these prefix bytes (i.e. partial AID matching) may be accepted in the following commands:</p> <ul style="list-style-type: none"> • INSTALL [for load] • INSTALL [for install] • INSTALL [for make selectable] • INSTALL [for install & make selectable] • INSTALL [for extradition] <p>For this command, only the Application AID shall be checked, not the one of the target SD.</p> <ul style="list-style-type: none"> • INSTALL [for registry update] <p>For this command, only the Application AID shall be checked, not the one of the target SD.</p> <p>If this tag is not present, the INSTALL command is not allowed. Several occurrences of this tag may be present to provide several patterns.</p>	C
'C5'	3	<p>Privileges that may be assigned to an Application created through this secure channel.</p> <p>Only the specified privileges may be assigned in the following commands:</p> <ul style="list-style-type: none"> • INSTALL [for install] • INSTALL [for make selectable] • INSTALL [for install & make selectable] <p>Notice however that a Security Domain may not grant more privileges (to another Security Domain) than it has itself.</p>	O		

Tag	Length	Value Description	MOC		
'EF'	Var	System specific parameters			
		Tag	Length	Value Description	MOC
		'82'	2 or 4	<p>Maximum cumulative granted volatile memory that may be defined for a Security Domain created through this secure channel.</p> <p>If this tag is present, then:</p> <ul style="list-style-type: none"> Any attempt to allocate a volatile CGM exceeding this value shall fail. Any attempt to install a new Security Domain shall include the specification of a volatile CGM value, unless such a value has already been defined for a Security Domain higher in the hierarchy of the new Security Domain. 	O
		'83'	2 or 4	<p>Maximum cumulative granted non-volatile memory that may be defined for a Security Domain created through this secure channel.</p> <p>If this tag is present, then:</p> <ul style="list-style-type: none"> Any attempt to allocate a non-volatile CGM exceeding this value shall fail. Any attempt to install a new Security Domain shall include the specification of a non-volatile CGM value, unless such a value has already been defined for a Security Domain higher in the hierarchy of the new Security Domain. 	O
		'8B'	Var	<p>List of allowed DGI numbers in STORE DATA.</p> <p>The list is formatted as a sequence of the DGI tags.</p> <p>If this tag is not present, the DGI variant of STORE DATA is not allowed.</p> <p>Notice however that this rule only applies to a STORE DATA command targeting a Security Domain. No restrictions apply to a STORE DATA command received by a regular Application (directly or forwarded by its associated Security Domain).</p>	O
'8C'	Var	<p>List of allowed TLV tags in STORE DATA.</p> <p>The list is formatted as a sequence of the BER-TLV tags.</p> <p>If this tag is not present, the TLV variant of STORE DATA is not allowed.</p> <p>Notice however that this rule only applies to a STORE DATA command targeting a Security Domain. No restrictions apply to a STORE DATA command received by a regular Application (directly or forwarded by its associated Security Domain).</p>	O		

Tag	Length	Value	Description	MOC
		'8E'	5-16 AID prefix bytes used to control which Application or ELF may be deleted. Only Application and ELF AID values matching these prefix bytes (i.e. partial AID matching) may be accepted. If this tag is not present, the DELETE command is not allowed. Several occurrences of this tag may be present to provide several patterns. The cumulative DELETE command (P2.b7=1) as defined in Amendment C: Contactless Services [Amd C] shall be supported if the SD processing the SCP11c session has the Global Delete privilege.	O
		'8F'	5-16 AID prefix bytes used to control which Executable Load File may be upgraded. Only ELF AID values matching these prefix bytes (i.e. partial AID matching) may be accepted. If this tag is not present, the MANAGE ELF UPGRADE command is not allowed. Several occurrences of this tag may be present to provide several patterns.	O

Security Domains shall ignore unknown tags within tags 'E3' and 'EF' to allow for future extensibility.

Note: If SCP11c is only used to protect scenario #1 of [Amd A], the contents of tag '8B' should be '00 DE 00 A6' to authorize storing an AP certificate and trigger key generation. If SCP11c is only used to protect scenario #3 of [Amd A], the contents of tag '8B' should be '7F 49 00 A6' to authorize the intended key generation.

If allowed, the commands shall be executed as described in [GPCS].

If the script attempts to perform an operation that is not allowed, the command triggering this operation shall be rejected with a status word of '6985', the SCP11c session shall be terminated, and all objects with a "pending completion" mark shall be deleted (see section B.3).

The following table provides another view of this authorization mechanism sorted by APDU command:

Table B-3: Possible Authorizations per APDU Command

Command	Acceptance Condition
DELETE [AID]	Only allowed if tag '8E' is present in the OCE certificate.
DELETE [key(s)]	Not allowed
Cumulative DELETE	Only allowed if the conditions for the DELETE command are satisfied and the SD processing the SCP11c session has the Global Delete privilege. See section B.6.
GET DATA	Always allowed
GET STATUS	Not allowed
INSTALL	Only allowed if tag '8A' is present in the OCE certificate. The following additional restrictions apply to specific INSTALL commands:
[for registry update]	Only privileges listed in tag 'C5' may be assigned
[for personalization]	(no additional rule)
[for extradition]	(no additional rule)
[for install]	Only privileges listed in tag 'C5' may be assigned
[for make selectable]	Only privileges listed in tag 'C5' may be assigned
[for install & make selectable]	Only privileges listed in tag 'C5' may be assigned
[for load]	(no additional rule)
LOAD	Only allowed if tag '8A' is present in the OCE certificate.
MANAGE ELF UPGRADE	Only allowed if tag '8F' is present in the OCE certificate.
PERFORM TRANSACTION OPERATION	Always allowed
PUT KEY	Not allowed
SET STATUS	Not allowed
SELECT	Always allowed, but will terminate the SCP11c session
STORE DATA	Only allowed if tag '8B' or '8C' is present in the OCE certificate. Only DGLs listed in tag '8B' or TLVs listed in tag '8C' shall be accepted.

B.3 Detection and Removal of Pending Card Contents

The execution of an SCP11c script may be interrupted by a power loss or may need to abort because a secure messaging error was detected. For this reason, a transaction mechanism is defined which may be used within an SCP11c session to prevent incomplete setup of card contents in case of interruption. This transaction mechanism works as follows:

- The PERFORM TRANSACTION COMMAND (see section B.4) shall be used to begin or commit a transaction.
- If a new ELF is loaded or a new Applet (or Security Domain) instance is installed during an ongoing transaction, it shall be marked internally as “pending completion”.
- All “pending completion” marks shall be cleared when a PERFORM TRANSACTION COMMAND [commit] command is received.
- If any of the following events occurs during an ongoing transaction, all objects having a “pending completion” mark shall be deleted as soon as possible (and before processing the next APDU command):
 - The SCP11c session is closed. See [GPCS] for Secure Channel Termination.
 - The card is powered down or the I/O interface (contact, contactless) on which the SCP11c session was established is reset.
 - An attempt to open a new SCP11c session is made on the same logical channel.

Note 1: *This transaction mechanism only provides limited guarantees that the card will be in the same state as before in case of rollback. For example, any modification performed on some data that does not belong to an Applet instance created by the script may persist after the rollback.*

Note 2: *A script command failing for a reason other than a secure messaging error is not defined in this mechanism as an event that shall trigger the abortion of the script. Moreover, the SCP11c SD processing the script also doesn't have any clue as to whether it should abort the script execution in such cases. If aborting the script execution is the desired course of action, the off-card entity providing the script may trigger the abortion by closing the SCP11c session, which may be simply achieved by, for example, selecting another Application or re-selecting the SCP11c SD on the same logical channel or closing the logical channel on which the latter is selected.*

B.4 PERFORM TRANSACTION OPERATION Command

This section specifies the command which is used to begin or commit a transaction.

B.4.1 Definition and Scope

The PERFORM TRANSACTION OPERATION command begins or commits a transaction. An SCP11c script may use this command to benefit from automatic rollbacks in case of interrupted script executions. This command shall be rejected if received out of an SCP11c session.

B.4.2 Command Message

The PERFORM TRANSACTION OPERATION command message is coded according to the following table.

Table B-4: PERFORM TRANSACTION OPERATION Command Message

Code	Value	Meaning
CLA	'84' - '87' or 'E0' - 'EF'	See [GPCS] section 11.1.4.
INS	'12'	PERFORM TRANSACTION OPERATION
P1	'00'	
P2	'80'	BEGIN TRANSACTION
	'81'	COMMIT TRANSACTION
Lc		Not present
Data		Not present
Le		Not present

- If a Security Domain receives a BEGIN TRANSACTION command, the Security Domain shall start marking newly installed objects as “pending completion” (see section B.3).
 - If two consecutive BEGIN TRANSACTION commands are received with no COMMIT TRANSACTION command in between, the SCP11c session shall be terminated and all objects with a “pending completion” mark shall be deleted.
- If a Security Domain receives a COMMIT TRANSACTION command, the Security Domain shall remove the “pending completion” mark from all objects created by commands received through SCP11c.

B.4.3 Response Message

The data field of the response message is empty.

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 B-5: PERFORM TRANSACTION OPERATION Error Conditions

SW1	SW2	Meaning
'69'	'85'	A COMMIT TRANSACTION was received without a prior BEGIN TRANSACTION

B.5 Self-Deletion via DELETE

This specification defines a new mechanism which allows a Security Domain to delete itself and all dependent packages and instances. At the end of this process, all resources associated with this SD will be deleted.

A DELETE command shall be recognized as a self-deletion command if the following conditions are met:

- The AID to be deleted (contents of tag '4F') is identical to the AID of the SD instance that receives the command and
- The reference control parameter P2 is set accordingly (P2.b6=1).

The following table shows the values for the reference control parameter P2 used to indicate self-deletion:

Table B-6: Values for Reference Control Parameter P2

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	-	-	-	-	-	-	-	
1	-	-	-	-	-	-	-	Delete object and related object
0	1	-	-	-	-	-	-	Delete a root Security Domain and all associated Applications
0	0	1	-	-	-	-	-	Self-Delete Security Domain
-	-	-	X	X	X	X	X	RFU

Self-deletion is only allowed if the following conditions are met:

- The current Security Level is AUTHENTICATED (or higher); and
- The Security Domain was previously created during an SCP11c session.

When a self-deletion command is received, the following process shall be executed:

- If the SD or any of its dependent SDs or Applications is selected on another logical channel, the command shall be rejected.
- The SD shall:
 - Check that the current Security Level is AUTHENTICATED (or higher).
- The SD shall request the OPEN to apply Cumulative Deletion (as defined in [Amd C]) to itself. Doing so, the OPEN shall:
 - Check that the card Life Cycle State is neither CARD LOCKED nor TERMINATED.
 - Check that the SD was created during an SCP11c session.
 - Deselect the SD (note that this has the effect of closing the secure channel).
 - Perform Cumulative Deletion of the SD.
 - If Cumulative Deletion was successful, return a status word of '9000'. Otherwise, return an error status word.

B.6 Cumulative DELETE by Parent SD

The provisions of [Amd C] section 10 of (Cumulative DELETE) shall apply if the following conditions are met:

- The Security Domain that processes the SCP11c session has the Global Delete privilege.
- The AID to be deleted refers to a Security Domain.
- Tag '8E' allows DELETE.
- The reference control parameter P2 of the DELETE command is set to “Delete a root Security Domain and all associated Applications” (see Table B-6).