GlobalPlatform Card

Security Upgrade for Card Content Management

Card Specification v 2.2 – Amendment E

Version 1.0.0.3

Public Review

May 2014

Document Reference: GPC_SPE_042

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

Contents

Figures

Tables

1 Introduction

The security in GlobalPlatform Card Specification [GPCS] and amendments A, B, C, and D is based on several cryptographic primitives (e.g. TDEA, RSA, AES…). The purpose of this amendment is to expand those specifications to include new cryptographic schemes based on Elliptic Curve Cryptography (ECC) and upgraded cryptographic schemes for RSA.

Configurations will define the selection of cipher suites that are mandatory or optional for cards in a specific market segment.

1.1 Audience

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

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

1.2 IPR Disclaimer

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

1.3 Normative References

Table 1-1: Normative References

Standard / Specification	Description	Ref
GlobalPlatform Card Specification	GlobalPlatform Card Specification v 2.2.1	[GPCS]
GPCS Amendment A	GlobalPlatform Card, Confidential Card Content Management, Card Specification v2.2 – Amendment A, v1.0.1	[Amd A]
GPCS Amendment D	GlobalPlatform Card Technology, Secure Channel Protocol 03, Card Specification v 2.2 – Amendment D, v1.1	[Amd D]
NIST SP 800-38B	Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, May 2005	[NIST 800-38B]
NIST SP 800-56A Revision 1	Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, March 2007	[NIST 800-56A]
NIST SP 800-57 Part 1 revised2	Recommendation for Key Management – Part 1: General (Revised), March 2007	[NIST 800-57]

Standard / Specification	Description	Ref
NIST SP 800-90 revised	Recommendation for Random Number Generation Using Deterministic Random Bit Generators, March 2007	[NIST 800-90]
ISO/IEC 7812	Identification cards – Identification of Issuers	[ISO 7812]
ISO/IEC 7816-4:2005	Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange	[ISO 7816-4]
ISO 9797-1	Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher	[ISO 9797-1]
ISO/IEC 14888-3:2006	Information technology – Security techniques – Digital signatures with appendix – Part 3: Discrete logarithm based mechanisms	[ISO 14888]
ANSI X9.62:2005	Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA)	[X9.62]
FIPS PUB 186-3	Digital Signature Standard (DSS)	[FIPS 186]
BSI TR-02102, Version 1.0	BSI Technische Richtlinie TR-02102: Kryptographische Verfahren: Empfehlungen und Schlüssellängen	[TR 02102]
BSI TR-03111, Version 1.11	BSI Technical Guideline TR-03111: Elliptic Curve Cryptography	[TR 03111]
RFC 5639	Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation	[RFC 5639]
PKCS #1	PKCS #1 v2.1: RSA Cryptography Standard, RSA Laboratories, June 14, 2002	[PKCS 1]

1.4 Terminology and Definitions

Technical terms used in this amendment are defined in [GPCS].

1.5 Abbreviations and Notations

Table 1-2: Abbreviations

Abbreviation	Meaning
AES	Advanced Encryption Standard
AID	Application IDentifier
AP	Application Provider
APDU	Application Protocol Data Unit
APSD	Application Provider Security Domain
BSI	Bundesamt für Sicherheit in der Informationstechnik
CASD	Controlling Authority Security Domain

Abbreviation	Meaning
CBC	Cipher Block Chaining
CERT.CASD.ECKA	Certificate of the CASD
CMAC	Cipher-based MAC
CRT	Control Reference Template
CT	CRT for Confidentiality
DAP	Data Authentication Pattern
DEK	Data Encryption Key
DGI	Data Grouping Identifier
DR	Derivation Random
DSS	Digital Signature Standard
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECKA	Elliptic Curve Key Agreement Algorithm
ECKA-EG	ElGamal ECKA
ePK.AP.ECKA	ephemeral Public Key of the AP
eSK.AP.ECKA	ephemeral Private Key of the AP
FIPS	Federal Information Processing Standard
GF(p)	Galois Field or Finite Field with p elements – p being a prime
ICV	Initial Chaining Vector
KAT	CRT for Key Agreement
KIc	SCP80 Ciphering key
KID	SCP80 authentication key
LFDB	Load File Data Block
LFDBH	Load File Data Block Hash
LPO	Link Platform Operator
MAC	Message Authentication Code
MGF	Mask Generation Function
NIST	National Institute of Standards and Technology
OAEP	Optimal Asymmetric Encryption Padding
OTA	Over The Air
PK.CASD.ECKA	Public Key of the CASD
PKCS	Public Key Cryptography Standards
PSS	Probabilistic Signature Scheme
RFU	Reserved for Future Use

Abbreviation	Meaning
RSA	Rivest / Shamir / Adleman asymmetric algorithm
RSAES	RSA Encryption Scheme
RSASSA	RSA Signature Scheme with Appendix
S-ENC	Secure channel key for Encryption
S-MAC	Secure channel key for MACing
SCP	Secure Channel Protocol
SD	Security Domain
SHA-1	Secure Hash Algorithm 1 (digest size 160 bits)
SHA-xxx	Secure Hash Algorithm with digest size xxx bits
ShS	Shared Secret
SK.CASD.ECKA	Private Key of the CASD
TLS	Transport Layer Security
TLV	Data structure containing of Tag, Length Value fields
TDEA	Triple DEA (Data Encryption Algorithm)

1.6 Revision History

Table 1-3: Revision History

Date	Version	Description
November 2011	1.0	Initial release.
April 2014	1.0.0.2	• Added section 3.3 “Key Usage Qualifier” describing a new qualifier value for Key Agreement. • In section 4.5, clarified the purpose of global and local key parameter references for preloaded ECC curve parameters. Only the ISD may store global key parameter references. • In section 4.6, clarified the format used to load ECC keys using the PUT KEY command. • In section 4.8, added many clarifications for “scenario #3” regarding: o the format of DGI '00A6' which triggers key set generation; o how to generate key sets having keys of different types and lengths; o CASD personalization data (including CASD certificate format); • In section 4.9, added clarifications on how to apply security on chained commands and chained responses. • Added section 4.10 “Key Information Template ('E0') for ECC Keys” • In section 6.1, clarified conditions for the presence of sub-tags within Card Capability Information. • In section 6.2, removed the ability to retrieve CASD Capability Information as an individual data with the GET DATA command. CASD Capability Information can still be retrieved as part of tag '66' (CASD Management Data).
May 2014	1.0.0.3	Public Review.

2 Use Cases and Requirements

Asymmetric Cryptography in [GPCS] is based on RSA with SHA-1 for hashing and the padding scheme from PKCS #1 v1.5. Neither SHA-1 nor the padding scheme is recommended any longer by security organizations like NIST or BSI (see [NIST 800-57] or [TR 02102]).

The purpose of this document is to provide alternative asymmetric security options in line with current recommendations.

To enable compact data structures, ECC is specified for public key cryptography.

Enhanced security mechanisms are provided for:

- Tokens and receipts, which are used in Delegated Management;
- DAPs, which are used for the protection of load files;
- and the Confidential Setup of Secure Channel Keys.

To distinguish the RSA based security mechanisms specified in this document from the mechanisms specified in [GPCS], the mechanisms in this document are referenced as “Variant 2 mechanisms”.

3 Specification Amendments

3.1 ECC Algorithms

3.1.1 Domain Parameters and Key Length

Elliptic Curve Cryptography over prime fields GF(p) shall be used for the purpose of this amendment.

Standardized Domain Parameters are recommended to be used. Such Parameters can be found in:

- Digital Signature Standard (DSS) [FIPS 186], recommended by NIST, or
- Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation [RFC 5639], recommended by BSI.

The following table lists recommended curves for different ECC key lengths.

Table 3-1: ECC Key Length and Recommended Curves

ECC Key Length	Curve Specified in [FIPS 186]	Curve Specified in [RFC 5639]
256	P-256	brainpoolP256r1 brainpoolP256t1
384	P-384	brainpoolP384r1 brainpoolP384t1
512	-	brainpoolP512r1 brainpoolP512t1
521	P-521	-

3.1.2 ECC Key Type

Table 11-16 of [GPCS] shall be extended with the following entries:

Table 3-2: Key Type Coding

Value	Meaning
'B0'	ECC public key
'B1'	ECC private key
'B2'	ECC field parameter P (field specification)
'B3'	ECC field parameter A (first coefficient)
'B4'	ECC field parameter B (second coefficient)
'B5'	ECC field parameter G (generator)
'B6'	ECC field parameter N (order of generator)
'B7'	ECC field parameter k (cofactor of order of generator)
'F0'	Key parameter reference

3.1.3 ECDSA

ECDSA (sometimes named EC-DSA) is specified in ANSI X9.62 [X9.62] standard.

From an input message M, ECDSA produces a signature (r,s).

ECDSA requires a hash function. Unless defined otherwise, the security strength of the hash function used shall meet or exceed the security strength associated with the order of the key according to [FIPS 186].

Table 3-3 lists the Hash algorithms that shall be used depending on specific ECC key lengths.

Table 3-3: Hash Algorithms for ECDSA

ECC Key Length (in bits)	Hash Algorithm
256-383	SHA-256
384-511	SHA-384
512+	SHA-512

The signature shall be coded in plain format as specified in [TR 03111], i.e. it is the concatenation of the byte string representation of r and s. Thus the signature will have a fixed length of twice the order length.

The ECDSA signature algorithm requires a random value as input. To protect against attacks, a high quality random number generator is required for the entity generating the signature. Recommendations for appropriate random number generators are given in [TR 02102] and [NIST 800-90].

3.1.4 ECKA

An Elliptic Curve Key Agreement Algorithm (ECKA) is used in this specification for the confidential setup of a new key set in an SD. A description of such schemes can be found e.g. in [TR 03111].

ECKA used in this specification shall follow the definition for the ElGamal Key Agreement ECKA-EG in [TR 03111], which uses one static and one ephemeral key pair. This scheme is equivalent to the scheme named “One-pass Diffie-Hellmann, C(1, 1, ECC CDH)” in [NIST 800-56A]. The recommendation in [NIST 800-56A] should be taken into account in an implementation.

The static or ephemeral public key shall be coded in plain format as specified for signatures in [TR 03111], i.e. it is the concatenation of the byte string representation of r and s. Thus the keys will have a fixed length of twice the order length.

3.1.5 Key Derivation

The shared secret ShS generated by ECKA-EG 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 key set or the base key of a security domain are derived from an initial secret as defined in [TR 03111] for the “X9.63 Key Derivation Function”. On request, this key derivation may include additional entropy (a random number DR) generated on the card, which then becomes part of the “SharedInfo” of the key derivation algorithm.

3.2 RSA Algorithms (Variant 2)

[GPCS] already contains cryptographic schemes based on RSA. This specification defines new schemes for RSA keys using different signing, encryption and padding schemes.

3.2.1 Key Length

The new schemes defined in this specification shall be used with RSA keys longer than 1024 bits. It is recommended that the key length is an integer multiple of 32.

The RSA keys follow the Key Type Codes specified in Table 11-16 of [GPCS].

3.2.2 Signature Schemes

The following signature scheme shall be used for RSA signatures:

- RSASSA-PSS as defined in PKCS#1 [PKCS 1].

The following RSASSA-PSS scheme parameters shall be used for hash algorithm, mask generation function, salt length and trailer field:

- Hash algorithm: SHA-256.
- MGF1 as defined in [PKCS 1] shall be used as mask generation function and the underlying hash function shall be SHA-256.
- PKCS#1 default salt length shall be the octet length of the hash value.
- PKCS#1 defined 'BC' shall be used as trailer field.

3.2.3 Encryption Schemes

The following RSA encryption scheme shall be used:

- RSAES-OAEP as defined in [PKCS 1].

The following RSA-OAEP scheme parameters shall be used for hash algorithm, mask generation function and (optional) label:

- Hash algorithm: SHA-256.
- MGF1 as defined in [PKCS 1] shall be used as mask generation function and the underlying hash function shall be SHA-256.
- An empty label parameter shall be used.

3.3 Key Usage Qualifier

The definition of Key Usage Qualifier (see section 11.1.9 of [GPCS]) is extended with one extra byte as follows:

Table 3-4: Key Usage Qualifier (Byte 1) [Unchanged]

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	–	–	–	–	–	–	–	Verification (DST, CCT, CAT), Encipherment (CT)
–	1	–	–	–	–	–	–	Computation (DST, CCT, CAT), Decipherment (CT)
–	–	1	–	–	–	–	–	Secure messaging in response data fields (CT, CCT)
–	–	–	1	–	–	–	–	Secure messaging in command data fields (CT, CCT)
–	–	–	–	1	–	–	–	Confidentiality (CT)
–	–	–	–	–	1	–	–	Cryptographic Checksum (CCT)
–	–	–	–	–	–	1	–	Digital Signature (DST)
–	–	–	–	–	–	–	1	Cryptographic Authorization (CAT)

Table 3-5: Key Usage Qualifier (Byte 2) [New]

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	–	–	–	–	–	–	–	Key Agreement (KAT)
–	–	–	–	–	–	–	–	(RFU)

4 Card Content Management Usage

This section defines the provisions if ECC schemes or RSA Variant 2 schemes are used for card content management activities.

4.1 ECC based DAPs, Tokens, and Receipts

If a Load File Data Block Signature (DAP) is generated with an ECC key, the ECDSA signature scheme defined in section 3.1.3 shall be used in the Signature Calculation operation specified in section C.3 of [GPCS].

ECC based tokens shall be generated using the ECDSA signature scheme defined in section 3.1.3. The data elements included in the respective token as specified in [GPCS] shall be used as input message M. The signature shall constitute the token to be included in the Command Data Field.

ECC based receipts shall be generated using the ECDSA signature scheme defined in section 3.1.3. The data elements included in the respective receipt as specified in [GPCS] shall be used as input message M. The signature shall constitute the receipt to be included in the Response Data Field.

4.2 RSA based DAPs, Tokens, and Receipts (Variant 2)

For RSA Variant 2 signatures for DAPs, Tokens and Receipts, the signatures defined in section B.3, Public Key Cryptography Scheme 1 (PKCS#1), of [GPCS] shall be replaced by the signatures defined in section 3.2.2 of this amendment.

4.3 Encrypted Load File with optional ICV

Table 11-58 of [GPCS] is replaced by the table below in order to introduce encrypted Load Files with optional ICVs:

Table 4-1: New Load File Data Block

Tag	Length	Name	Presence
'E2'	1-n	DAP Block	Conditional
'4F'	5-16	Security Domain AID	Conditional
'C3'	1-n	Load File Data Block Signature	Conditional
:	:	:	
:	:	:	
'E2'	1-n	DAP Block	Conditional
'4F'	5-16	Security Domain AID	Conditional
'C3'	1-n	Load File Data Block Signature	Conditional
'C4'	1-n	Load File Data Block	Conditional
'D3'	8 or 16	ICV	Optional
'D4'	1-n	Encrypted Load File Data Block	Conditional

The decryption of the Encrypted Load File Data block is performed by the Security Domain that is to be associated with the load file.

The encryption mechanism for the Encrypted Load File Data Block will be based on:

- Encryption algorithm corresponding to the key type coding of encryption key (i.e. DES or AES, see section 11.1.8 of [GPCS] – Key Type Coding).
- The load file shall be padded according to section B.4 of [GPCS].
- CBC encryption shall be used. If tag 'D3' is missing, a zero ICV shall be used. Else the ICV given in tag 'D3' shall be used which should be a random value. The length shall be the block size of the encryption algorithm.

4.4 Load File Data Block Hash

The Load File Data Block Hash is used to protect the integrity of the Load File.

On any particular card implementation, a unique hash algorithm shall be selected and used to generate the Load File Data Block Hash (LFDBH).

Several mechanisms require the presence of a LFDBH: DAPs, load file data block encryption or the presence of a Load Token.

For each mechanism, acceptable hash algorithms for the LFDBH are defined in the table below depending on the cryptographic strength of the mechanism. The hash algorithm used for generating the LFDBH shall fulfill the requirements from all mechanisms used on the card.

Table 4-2: Hash Selection for LFDBH

Hash Algorithm Used by DAP or Load Token	Algorithm for Load File Data Block Encryption	Acceptable Hash Algorithm(s) for LFDBH
SHA-1	Triple DES with 16 byte key	SHA-1 or SHA-256 or SHA-384 or SHA-512
SHA-256	AES-128	SHA-256 or SHA-384 or SHA-512
SHA-384	AES-192	SHA-384 or SHA-512
SHA-512	AES-256	SHA-512

4.5 Preloaded ECC Curve Parameters

A card may have one or multiple sets of preloaded ECC curve parameters. A set of curve parameters is identified by a key parameter reference that is assigned at the time curve parameters are loaded. It is then possible to refer to a set of curve parameters, with a particular key parameter reference, when loading a new public or private ECC key, which avoids transferring curve parameters with each new key.

Key parameter references are coded on 1 or 2 bytes, bit b8 of the first byte indicating a coding on 2 bytes. Two kinds of key parameter references (respectively curve parameters) are defined:

- Global key parameter references (respectively curve parameters) can only be stored by the ISD and, when loading a new key, can be referenced from any Security Domain on the card.
- Local key parameter references (respectively curve parameters) can be stored by any Security Domain on the card and, when loading a new key, can be referenced from this Security Domain or any Security Domain in its sub-hierarchy.

Table 4-3 describes reserved ranges of values for each kind of key parameter reference. A range is reserved for global key parameter references defined by GlobalPlatform (RFU), another range is reserved for proprietary global key parameter references, and another one is reserved for proprietary local key parameter references. In this case, “proprietary” either means that no standard reference has been assigned to it yet by GlobalPlatform, or that a standard set of parameters is loaded, at the discretion of a Security Domain owner, under a local key parameter reference (e.g. because not preloaded by the card issuer).

Table 4-3: Key Parameter Reference Values

Curve	Description	Key Parameter Reference Value
P-256	as specified in [FIPS 186]	'00'
P-384		'01'
P-521		'02'
brainpoolP256r1	as specified in [RFC 5639]	'03'
brainpoolP256t1		'04'
brainpoolP384r1		'05'
brainpoolP384t1		'06'
brainpoolP512r1		'07'
brainpoolP512t1		'08'
RFU		'09' to '3F' and '80 00' to 'BF FF'
Reserved for proprietary global key parameter references		'40' to '5F' and 'C0 00' to 'DF FF'
Reserved for proprietary local key parameter references		'60' to '7F' and 'E0 00' to 'FF FF'

4.6 PUT KEY (ECC Key)

To allow multiple key components to be included in one PUT KEY command, Table 11-68 of [GPCS] shall be replaced by the following table:

Table 4-4: Key Data Field – Basic

Name	Length	Value	Presence
Key type of the first or only key component	1	'00' - 'FE' – see Table 3-2: Key Type Coding	Mandatory
Length of first or only key component	1-3	'01' - '80', or '81 80' - '81 FF', or '82 01 00' - '82 FF FF'	Mandatory
Key or first key component data	1-n	'xxxx...'	Mandatory
…	…	…	
Key type of the last key component, if more than one	1	'00' - 'FE' - see Table 3-2: Key Type Coding	Conditional
Length of last key component	1-3	'01' - '80', or '81 80' - '81 FF', or '82 01 00' - '82 FF FF'	Conditional
Last key component data	1-n	'xxxx...'	Conditional
Length of key check value	1	'00' - '7F'	Mandatory
Key check value	0-n	'xxxx...'	Conditional

> ***Note:*** *As standardized key types are coded with values greater than '7F', the length field for the key check value can unambiguously be detected for such keys.*

The key component value shall be formatted using uncompressed encoding as specified in section 3.1.1 of [TR 03111], with most significant byte coming first (hence the value shall start with the coding identifier byte '04').

If the key component value must be encrypted, the key component data contains a length field indicating the length of the key component value followed by the encrypted key component value, as defined in GPCS Amendment D [Amd D] for “PUT KEY Command (AES Key-DEK)”. This allows loading encrypted keys of any length.

All ECC keys or field parameters to be loaded shall be contained in one key data field of the PUT KEY command. The command itself may be chained, see section 4.9.1.

No key check mechanism for ECC keys is defined, i.e. the length of the key check value shall be '00'.

For loading of ECC curve parameters into a security domain, PUT KEY shall contain the following key components:

Table 4-5: Loading of ECC Curve Parameters

Key Type	Key Component	Presence	Data Content Encrypted
'F0'	Key parameter reference	Mandatory	No
'B2'	ECC field parameter P (field specification)	Mandatory	No
'B3'	ECC field parameter A (first coefficient)	Mandatory	No
'B4'	ECC field parameter B (second coefficient)	Mandatory	No
'B5'	ECC field parameter G (generator)	Mandatory	No
'B6'	ECC field parameter N (order of generator)	Mandatory	No
'B7'	ECC field parameter k (cofactor of order of generator)	Optional	No

For loading of a public or a private ECC key that reference global or local ECC curve parameters, PUT KEY shall contain the following key components:

Table 4-6: Loading of ECC Key with Parameter Reference

Key Type	Key Component	Presence	Data Content Encrypted
'B0' or 'B1'	ECC public or private key	Mandatory	No for public / yes for private key
'F0'	Key parameter reference	Mandatory	No

For loading of a public or a private ECC with its own ECC curve parameters, PUT KEY shall contain the following key components:

Table 4-7: Loading of ECC Key with Own Parameters

Key Type	Key Component	Presence	Data Content Encrypted
'B0' or 'B1'	ECC public or private key	Mandatory	No for public / yes for private key
'B2'	ECC field parameter P (field specification)	Mandatory	No
'B3'	ECC field parameter A (first coefficient)	Mandatory	No
'B4'	ECC field parameter B (second coefficient)	Mandatory	No
'B5'	ECC field parameter G (generator)	Mandatory	No
'B6'	ECC field parameter N (order of generator)	Mandatory	No
'B7'	ECC field parameter k (cofactor of order of generator)	Optional	No

4.7 STORE DATA (ECC Key)

The STORE DATA command shall be coded as a Case 3 command. Each DGI described in the following sections shall be sent in a single STORE DATA command, i.e. a DGI shall not be split over multiple APDUs.

A Key Control Reference Template is used to describe curve parameters and keys sent to a Security Domain. The DGI for the Key Control Reference Template is defined as follows:

Table 4-8: DGI for Key Information Data

DGI	Length	Data Content	Data Content Encrypted
'00B9'	Var	Key Information Data	No

DGI '00B9' shall be followed by DGIs providing the values of ECC curve parameters [TR 03111] or keys. Their data format shall be:

1) Uncompressed encoding as specified in section 3.1.1 of [TR 03111], with most significant byte coming first;
2) Fixed length related to the byte length of the key, zero-padding added before the most significant bit as needed. This does not apply to the cofactor of order of generator, which shall be encoded with the smallest number of bytes. Note that a 521 bit key would be encoded on 66 bytes.
3) For all encrypted data grouping content defined in this section, padding shall be done as follows: If the length of the sensitive data is not an integer multiple of the block length of the encryption algorithm used by the DEK key (e.g. 8 bytes for DES encryption, 16 bytes for AES encryption), padding of arbitrary bytes shall be appended prior to encryption to fill the last block.
4) The encryption and decryption of the DGI's contents shall be performed using the Data Encryption Key (DEK) and the algorithm supported by the Secure Channel Protocol for sensitive data encryption/decryption.

4.7.1 DGIs for the ECC Curve Parameters

The data content of DGI '00B9' (Key Control Reference Template) for ECC curve parameters is defined in the following table:

Table 4-9: Data Content for DGI '00B9' – ECC Curve Parameters

Tag	Length	Description	Presence
'B9'	Var	CRT tag (CT)	Conditional
'80'	'01'	Key Type = 'F0' – Key parameter reference	Conditional
'85'	'01' or '02'	Key parameter reference value (local reference, value range see Table 4-3)	Conditional
'B9'	Var	CRT tag (CT)	Mandatory
'80'	'01'	Key Type = 'B2' – P (field specification)	Mandatory
'81'	'01'	Curve parameter length, in bytes (unsigned integer value)	Mandatory
'B9'	Var	CRT tag (CT)	Mandatory
'80'	'01'	Key Type = 'B3' – A (first coefficient)	Mandatory
'81'	'01'	Curve parameter length, in bytes (unsigned integer value)	Mandatory
'B9'	Var	CRT tag (CT)	Mandatory
'80'	'01'	Key Type = 'B4' – B (second coefficient)	Mandatory
'81'	'01'	Curve parameter length, in bytes (unsigned integer value)	Mandatory
'B9'	Var	CRT tag (CT)	Mandatory
'80'	'01'	Key Type = 'B5' – G (generator)	Mandatory
'81'	'01'	Curve parameter length, in bytes (unsigned integer value)	Mandatory
'B9'	Var	CRT tag (CT)	Mandatory
'80'	'01'	Key Type = 'B6' – N (order of generator)	Mandatory
'81'	'01'	Curve parameter length, in bytes (unsigned integer value)	Mandatory
'B9'	Var	CRT tag (CT)	Optional
'80'	'01'	Key Type = 'B7' – k (cofactor of order of generator)	Optional
'81'	'01'	Curve parameter length, in bytes (unsigned integer value)	Optional

When loading ECC curve parameters that are intended to be referenced, the CRT with the Key Parameter Reference shall be present.

When loading ECC curve parameters that are directly related to one key only (see below), the CRT with the Key Parameter Reference shall be absent.

The following DGIs are used to populate the ECC curve parameters and shall immediately follow DGI '00B9':

Table 4-10: Data Grouping Identifiers for ECC Curve Parameters

DGI	Length	Data Content	Presence	Data Content Encrypted
'0030'	Variable	P (field specification)	Mandatory	No
'0031'	Variable	A (first coefficient)	Mandatory	No
'0032'	Variable	B (second coefficient)	Mandatory	No
'0033'	Variable	G (generator)	Mandatory	No
'0034'	Variable	N (order of generator)	Mandatory	No
'0035'	Variable	k (cofactor of order of generator); default value '01'	Optional	No

4.7.2 DGIs for the ECC Public Key

The data content of DGI '00B9' (Key Control Reference Template) for an ECC Public Key is defined in the following table:

Table 4-11: Data Content for DGI '00B9' – ECC Public Key

Tag	Length	Data Element	Presence
'B9'	Variable	CRT tag (CT)	Mandatory
'95'	'01'	Key Usage Qualifier values according to section 11.1.9 of [GPCS]	Mandatory
'80'	'01'	Key Type = 'B0' – ECC public key	Mandatory
'81'	'01' or '02'	Key Length in bytes (unsigned integer value)	Mandatory
'82'	'01'	Key Identifier	Mandatory
'83'	'01'	Key Version Number	Mandatory
'B9'	Var	CRT tag (CT)	Conditional
'80'	'01'	Key Type = 'F0' – Key parameter reference	Conditional
'85'	'01' or '02'	Key Parameter Reference Value	Conditional

The following Data Grouping Identifier is used to populate an ECC Public Key:

Table 4-12: Data Grouping Identifier for ECC Public Key

DGI	DGI Length	Data Content	Data Content Encrypted
'0036'	Variable	Q (public key)	No

When loading a key that shall use ECC curve parameters already present on the card, the CRT with the Key Parameter Reference shall be present. In case of a local reference, the curve parameters with the given reference found in the SD, or if missing there, found in the closest ascendant SD shall be used.

If the CRT with the Key Parameter Reference is absent, DGI '0036' shall be followed by the DGIs for curve parameters as defined in section 4.7.1. The CRT with the Key Parameter Reference shall also be absent within DGI '00B9' of the Curve Parameters.

4.7.3 DGIs for the ECC Private Key

The data content of DGI '00B9' (Key Control Reference Template) for an ECC Private Key is defined in the following table:

Table 4-13: Data Content for DGI '00B9' – ECC Private Key

Tag	Length	Data Element	Presence
'B9'	Variable	CRT tag (CT)	Mandatory
'95'	'01'	Key Usage Qualifier values according to section 11.1.9 of [GPCS]	Mandatory
'80'	'01'	Key Type = 'B1' – ECC private key	Mandatory
'81'	'01' or '02'	Key Length in bytes (unsigned integer value)	Mandatory
'82'	'01'	Key Identifier	Mandatory
'83'	'01'	Key Version Number	Mandatory
'B9'	Var	CRT tag (CT)	Conditional
'80'	'01'	Key Type = 'F0' – Key parameter reference	Conditional
'85'	'01' or '02'	Key Parameter Reference Value	Conditional

The following Data Grouping Identifier is used to populate an ECC Private Key:

Table 4-14: Data Grouping Identifier for ECC Private Key

DGI	DGI Length	Data Content	Data Content Encrypted
'8137'	Variable	d (private key)	Yes

When loading a key that shall use ECC curve parameters already present on the card, the CRT with the Key Parameter Reference shall be present. In case of a local reference, the curve parameters with the given reference found in the SD, or if missing there, found in the closest ascendant SD shall be used.

If the CRT with the Key Parameter Reference is absent, DGI '8137' shall be followed by the DGIs for curve parameters as defined in section 4.7.1. The CRT with the Key Parameter Reference shall also be absent within DGI '00B9' of the Curve Parameters.

4.8 Confidential Setup of Secure Channel Keys using ECKA

In addition to scenarios #1, #2.A, and #2.B which are based on the mechanisms of GPCS Amendment A [Amd A], this section defines a new scenario based on ECKA, which is named “Scenario #3” (see also section 6.2).

In order to enable this new scenario, a Controlling Authority Security Domain (CASD) shall be installed and personalized with an ECC Private Key (SK.CASD.ECKA to be used for ECKA-EG as specified in [TR 03111]) and the corresponding Public Key certificate CERT.CASD.ECKA. This static key pair allows the response from the card to be authenticated. CASD personalization data are described in section 4.8.2.

4.8.1 Confidential Setup of Secure Channel Keys using Scenario #3 (ECKA)

A STORE DATA command shall be sent to set up a Secure Channel key set, which can either be sent directly to the APSD or sent to its associated (parent) SD if preceded by an INSTALL [for personalization] command. A Secure Channel with at least MAC protection is required for these commands.

Parameters in the command control:

- if an additional derivation random DR shall be generated and used in the key derivation process, and
- if other key sets already stored by APSD shall be deleted.

Thus this mechanism allows the following:

- Confidential setup of an initial key set in an APSD via the associated SD.
- Confidential setup of an additional key set by the APSD itself.
- Confidential replacement of existing key sets.

The Application Provider (AP) generates an ephemeral key pair (i.e. a key pair used only for one action and deleted thereafter) eSK.AP.ECKA and ePK.AP.ECKA. The public key is provided to the card in the STORE DATA command.

The STORE DATA command shall contain DGI '00A6' with a value encoded as described in Table 4-15 if all generated keys shall have the same type and length, or Table 4-16 if generated keys shall have different types and/or different lengths.

Table 4-15: Content of DGI '00A6' to Generate Secure Channel Keys of Same Types and Lengths

Tag	Length	Data Element	Presence
'A6'	Variable	CRT tag (KAT)	Mandatory
'90'	2	Scenario identifier and parameters	Mandatory
'95'	'01'	Key Usage Qualifier '5C' (1 secure channel base key) '10' (all secure channel keys)(see [GPCS] Table 11-17)	Mandatory
'96'	'01'	Key Access according to [GPCS] Table 11-18	Optional
'80'	'01'	Key Type according to [GPCS] Table 11-16	Mandatory
'81'	'01'	Key Length (in bytes)	Mandatory
'82'	'01'	Key Identifier = '00' - '7F'	Optional
'83'	'01'	Key Version Number = '01' - '7F'	Optional
'91'	Variable	Initial value of sequence counter	Optional
'45'	1-n	Security Domain Image Number (SDIN)	Optional
'84'	1-n	HostID (shall only be present if scenario parameter b3 is set)	Conditional

Table 4-16: Content of DGI '00A6' to Generate Secure Channel Keys of Different Types or Lengths

Tag	Length	Data Element	Presence
'A6'	Variable	CRT tag (KAT)	Mandatory
'90'	2	Scenario identifier and parameters	Mandatory
'B9'	Variable		Conditional
'95'	'01'	Key Usage Qualifier o '3C' (Pre Shared Key for Secure Messaging) o '34' (C-MAC + R-MAC) o '38' (C-ENC + R-ENC) o 'C8' (C-DEK + R-DEK) (see [GPCS] Table 11-17)	Mandatory
'96'	'01'	Key Access according to [GPCS] Table 11-18	Optional
'80'	'01'	Key Type according to [GPCS] Table 11-16	Mandatory
'81'	'01'	Key Length (in bytes)	Mandatory
'82'	'01'	Key Identifier = '00' - '7F'	Optional
'83'	'01'	Key Version Number = '01' - '7F'	Optional
'B9'	Variable		Conditional
'95'	'01'	Key Usage Qualifier o '3C' (Pre Shared Key) o '34' (C-MAC + R-MAC) o '38' (C-ENC + R-ENC) o 'C8' (C-DEK + R-DEK) (see [GPCS] Table 11-17)	Mandatory
'96'	'01'	Key Access according to [GPCS] Table 11-18	Optional
'80'	'01'	Key Type according to [GPCS] Table 11-16	Mandatory
'81'	'01'	Key Length (in bytes)	Mandatory
'82'	'01'	Key Identifier = '00' - '7F'	Optional
'83'	'01'	Key Version Number = '01' - '7F'	Optional
…	…	…	…
'91'	Variable	Initial value of sequence counter	Optional
'45'	1-n	Security Domain Image Number (SDIN)	Optional
'84'	1-n	HostID (shall only be present if scenario parameter b3 is set)	Conditional

The scenario identifier shall be set to '03'.

The length of tag '91' depends on the Secure Channel Protocol that is expected to be used with the newly generated key set. If not present, the sequence counter (if any) shall be initialized to a value of 0.

The Secure Channel Protocol for which the Secure Channel keys are generated is implicitly known from the specified Key Version Number (tag '83') and the list of Secure Channel Protocols supported by the Security Domain processing the STORE DATA command. Such rules remain out of the scope of this document and shall be defined in configuration documents. If the specified key information (e.g. key type, key length, key identifier, etc.) or the number of keys is not consistent with the intended Secure Channel Protocol, an error shall be returned.

The scenario parameters are defined as follows:

Table 4-17: Parameters for Scenario #3

b8	b7	b6	b5	b4	b3	b2	b1	Description
–	–	–	–	–	–	–	1	Do not delete existing keys
–	–	–	–	–	–	1	–	Include DR in key derivation process
–	–	–	–	–	1	–	–	Include Host and Card ID in key derivation process
x	x	x	x	x	–	–	–	RFU (0)

If the indicated key set already exists, it shall be replaced; else a new key set shall be created.

If bit 1 is set (“Do not delete existing keys”), then all other keys in the SD shall remain unchanged. Else all keys in the SD except the newly generated key set shall be deleted. Using this option, it is possible for example to send several consecutive STORE DATA commands in order to generate several key sets.

If bit 3 is set (“Include Host and Card ID in key derivation process”) and tag '84' (Host ID) is not present within tag 'A6' (see Table 4-15), then an error shall be returned. Similarly, if bit 3 is not set and tag '84' (Host ID) is present within tag 'A6' (see Table 4-15), then an error shall be returned.

The card shall verify the values provided for scheme identifier, scheme parameters, key identifier and key type.

DGI '7F49' shall immediately follow DGI '00A6' and contains the ephemeral Public Key of the AP formatted using uncompressed encoding as specified in section 3.1.1 of [TR 03111], with most significant byte coming first (hence the value shall start with the coding identifier byte '04'). It is assumed that this ephemeral Public Key is based on the same ECC curve as the CASD Public Key (PK.CASD.ECKA).

Table 4-18: Data Grouping Identifier for Application Provider’s ECC Ephemeral Public Key

DGI	Length	Description
'7F49'	Var.	ePK.AP.ECKA

The card shall use ePK.AP.ECKA and SK.CASD.ECKA to generate the shared secret ShS according to section 3.1.4.

The card shall generate a Derivation Random (DR) if requested (see scenario parameter b2). The length of the generated DR depends on the length of the involved ECC keys as described in Table 4-19.

Table 4-19: Length of the Derivation Random

ECC key length in bits	Length of DR in bytes
256-383	16
384-511	24
512+	32

The card shall derive the secure channel keys from ShS and DR (if requested) according to section 3.1.5 and store them in the indicated location.

The concatenation of the following value shall be used for *SharedInfo*:

- for each key described under tag 'A6':
  - key usage qualifier (1 byte)
  - key type (1 byte)
  - key length (1 byte)
- if requested: DR (16, 24 or 32 bytes)
- if Host and Card ID are requested: HostID-LV, IIN-LV and CIN-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.

IIN-LV is the length and the value field of the Issuer Identification Number (see [GPCS]).

CIN-LV is the length and the value field of the Card Image Number (see [GPCS]).

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

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

In addition to the key(s) required for secure channel usage, a receipt key is used to calculate the receipt to be included in the response to the STORE DATA command. The type and length of the receipt key, as well as the signature algorithm used to generate the receipt, depend on the type and length of the generated secure channel key(s):

If the generated Secure Channel keys all share the same type and length (see Table 4-15), then:

- The receipt key shall be of the same type and length (L) as the generated Secure Channel keys.
- The receipt generation algorithm shall be the signature algorithm described in:
  - [ISO 9797-1] with details given in section B.1.2.2 of [GPCS], if the receipt key is a DES key.
  - [NIST 800-38B], if the receipt key is an AES key.
- *KeyData* shall be assigned sequentially to the receipt key and to each of the secure channel key(s) as described in Table 4-20.

Table 4-20: Assignment of *KeyData* for Keys of Same Types or Lengths

KeyData	**Key**
1 to L	receipt key
L+1 to 2L	1st secure channel key
2L+1 to 3L	2nd secure channel key
…	…

If the generated Secure Channel keys are of different types and/or lengths (see Table 4-16), then:

- The receipt key shall be a 16-byte AES key
- The receipt generation algorithm shall be the signature algorithm described in [NIST 800-38B].
- *KeyData* shall be assigned sequentially to the receipt key and to each of the secure channel key(s) as described in Table 4-21.

Table 4-21: Assignment of *KeyData* for Keys of Different Types or Lengths

KeyData	**Key**
1 to 16	receipt key
17 to 17+L1	1st secure channel key (length L1)
18+L1 to 18+L1+L2	2nd secure channel key (length L2)
…	…

If requested, the card shall delete all other keys in the APSD (see scenario parameter b1).

Finally, the card shall generate a receipt (using the receipt key and algorithm defined above) by calculating a signature across the data described in Table 4-22. The receipt key shall be deleted after calculating the receipt.

Table 4-22: Input Data for Receipt Calculation

Tag	Length	Data Element	Presence
'A6'	Variable	CRT TLV with all sub TLVs as provided in the STORE DATA command	Mandatory
'85'	Variable	DR	Conditional

The STORE DATA response shall contain the following data:

Table 4-23: Response Data for Scenario #3

Tag	Length	Data Element	Presence
'85'	Variable	DR	Conditional
'86'	Variable	Receipt	Mandatory

4.8.2 CASD Personalization Data for Scenario #3 (ECKA)

The CASD Certificate (CERT.CASD.ECKA) shall have the following structure (certificate without message recovery):

Table 4-24: CASD Certificate Structure (CERT.CASD.ECKA)

Tag	Length	Description	Presence
'7F21'	Var.	Certificate	Mandatory
'93'	1-16	Certificate Serial Number	Mandatory
'42'	Var.	CA Identifier	Mandatory
'5F20'	Var.	Subject Identifier	Mandatory
'95'	1	Key Usage ('00 80' key agreement) (see section 3.3)	Mandatory
'5F25'	4	Effective Date (YYYYMMDD, BCD format)	Optional
'5F24'	4	Expiration Date (YYYYMMDD, BCD format)	Mandatory
'45'	Var.	CA Security Domain Image Number	Mandatory
'53'	1-127	Discretionary Data (unspecified format)	Optional
'73'	1-127	Discretionary Data (BER-TLV encoded)	Optional
'7F49'	Var.	Public Key data object (see Table 4-26)	Mandatory
'5F37'	Var.	Signature	Mandatory

The CA Identifier (tag '42') identifies the owner of the CASD and shall be allocated according to the scheme described in [ISO 7812]. The Subject Identifier (tag '5F20') identifies the Secure Element uniquely.

Optional tags '53' and '73' are mutually exclusive. Tag '73' is intended for BER-TLV encoded discretionary data.

The following data shall be signed using the ECDSA algorithm (see section 3.1.3) to generate the signature of the CASD certificate (tag '5F37'):

Table 4-25: Data Signed to Generate the Signature of CERT.CASD.ECKA

Tag	Length	Description	Presence
'93'	1-16	Certificate Serial Number	Mandatory
'42'	Var.	CA Identifier	Mandatory
'5F20'	Var.	Subject Identifier	Mandatory
'95'	1	Key Usage	Mandatory
'5F25'	4	Effective Date (YYYYMMDD, BCD format)	Conditional
'5F24'	4	Expiration Date (YYYYMMDD, BCD format)	Mandatory
'45'	Var.	CA Security Domain Image Number	Mandatory
'53'	1-127	Discretionary Data (unspecified format)	Conditional
'73'	1-127	Discretionary Data (BER-TLV encoded)	Conditional
'7F49'	Var.	Public Key data object (see Table 4-26)	Mandatory

The format of the ECC Public Key data object (tag '7F49') is described in the following table:

Table 4-26: ECC Public Key Data Object Structure

Tag	Length	Description	Presence
'7F49'	Var.	Public Key data object	
'B0'	Var.	ECC Public key – Q	Mandatory
'F0'	1	Key Parameter Reference (Curve) '00': NIST P-256 '03': brainpoolP256r1	Mandatory

The corresponding ECC Private Key (SK.CASD.ECKA) shall be loaded in the CASD with a Key Version Number of '74' and a Key Identifier of '04'.

4.9 Long Parameter and Command and Response Data Fields

Because certificates, digital signatures and some data fields can be long, command and response chaining may be required.

4.9.1 Command Chaining

[GPCS] already defines suitable mechanisms for the chaining of command data fields for the DELETE, INSTALL and PUT KEY commands, using the P1 parameter.

The following precisions to [GPCS] shall apply for command chaining for these commands.

- All the commands of the sequence shall directly follow each other and have the same command header, except for Lc and the chaining information in P1. If this is not the case, a response of '6985' shall be returned and the chaining session shall be aborted.
- P1.b8 indicates if the command is expected to be followed by the next command of the chain or if the command is the last of a sequence of chained commands.
- The overall length supported with chaining is implementation dependent. If chaining is not supported or if the chaining limit has been reached, then a response of '6A86' shall be returned.
- The data fields of the sequence of commands shall be concatenated on the card to form a full set of data that may be consistently analyzed and processed. At any moment during the reception of the sequence of commands, an appropriate error code may be returned if an error is detected in the data field of a command, in which case the chaining session shall be aborted. However, an implementation is not required to check or process data before the entire sequence of commands has been received.
- For all commands of the sequence except for the last one, a response of '9000' with no additional response data shall be returned if no error has been detected, The regular response shall only be returned upon reception of the last command of the sequence.
- Security shall be applied to the command data before segmenting it into several commands. If a command includes a checksum (e.g. C-MAC) and the command data without the checksum exceeds 255 bytes, the calculation of the checksum shall be performed across the non-segmented command which is defined as follows:
  - Reference control parameters P1 and P2 shall be set as for the last command of the sequence of chained commands.
  - “Length of the following data fields” shall indicate the length of the data following. It shall be coded on 3 bytes with values from '00 01 00' to '00 FF FF'.
  - This shall be followed by the concatenation of the data fields of all commands of the chained sequence.

  The previous statement has an impact on how Secure Channel Protocols (e.g. SCP02 or SCP03) are applied to such a sequence of chained commands. In this case, the C-MAC shall be computed over the non-segmented command and shall be appended only at the end of the last command of the sequence. In the same manner, command data field encryption (if requested) shall be performed over the non-segmented command data field.

4.9.2 Response Chaining

For any command where the response exceeds 256 bytes, the chaining mechanism defined in ISO/IEC 7816-4 [ISO 7816-4], using the '61xx' status word and multiple GET RESPONSE commands, should be used. This has an impact on how Secure Channel Protocols (e.g. SCP02 or SCP03) shall be applied to response data. In this case, the R-MAC (if requested) shall be computed over the non-segmented response data and shall be appended only to the response of the last GET RESPONSE command of the sequence. In the same manner, response data field encryption (if requested) shall be performed over the non-segmented response data field.

4.9.3 Token Calculation for Chained Command Data Fields

Token calculation is specified in [GPCS] for command data up to 255 bytes prior to the token being added.

If a command includes a token and the command data without the token exceeds 255 bytes, the token shall be computed across the non-segmented command which is defined as follows:

- Reference control parameters P1 and P2 shall be set as for the last command of the sequence of chained commands.
- “Length of the following data fields” shall indicate the length of the data following prior to the token being added. It shall be coded on 3 bytes with values from '00 01 00' to '00 FF FF'.
- This shall be followed by the concatenation of the data fields of all commands of the chained sequence.

4.9.4 Long Parameter Fields

Section 11.1.5 of [GPCS] mentions that length fields for messages and data objects can be coded on up to 3 bytes, i.e. the following length values are permitted: '00' to '7F', '81 80' to '81 FF', and '82 01 00' to '82 FF FF'. (Only where explicitly indicated, '80' is also a valid coding.)

However, many length fields in tables given in Chapter 11 of [GPCS] are limited to 1 or 2 bytes, effectively limiting the value part of the message or object to 127 or 255 bytes.

Wherever it is required, e.g. for install parameters longer than 255 bytes, an implementation may support longer data objects and thus longer length fields than those given in Chapter 11 of [GPCS]. The same applies for the length fields of the data used in token and receipt calculations in sections C.4 and C.5 of [GPCS].

4.10 Key Information Template ('E0') for ECC Keys

The Key Information Template allows retrieving information about the cryptographic keys stored by a Security Domain. This template can be retrieved using a GET DATA command for tag 'E0'. The template itself contains one or several tags 'C0' (Key Information Data), each one describing a single key.

For an ECC key referencing preloaded curve parameters, the content of tag 'C0' shall be:

Table 4-27: Content of Tag 'E0' for an ECC Public or Private Key

Tag	Data Element		Presence
'E0'	Key Information Template		
…	…		…
'C0'	Key Information Data		
	Value	**Meaning**	
	'B0' or 'B1'	ECC Public Key ('B0') or Private Key ('B1')	Mandatory
	Variable	Length of ECC Public or Private Key (excluding coding identifier '04'): o '40' (ECC 256) o '60' (ECC 384) o '80' (ECC 512) o '82' (ECC 521)	Mandatory
	'F0'	Key Parameter Reference	Conditional
	Variable	Value of Key Parameter Reference	Conditional
…	…		…

Key Parameter Reference ('F0') shall be present if the ECC key was loaded with a reference to preloaded ECC Curve Parameters.

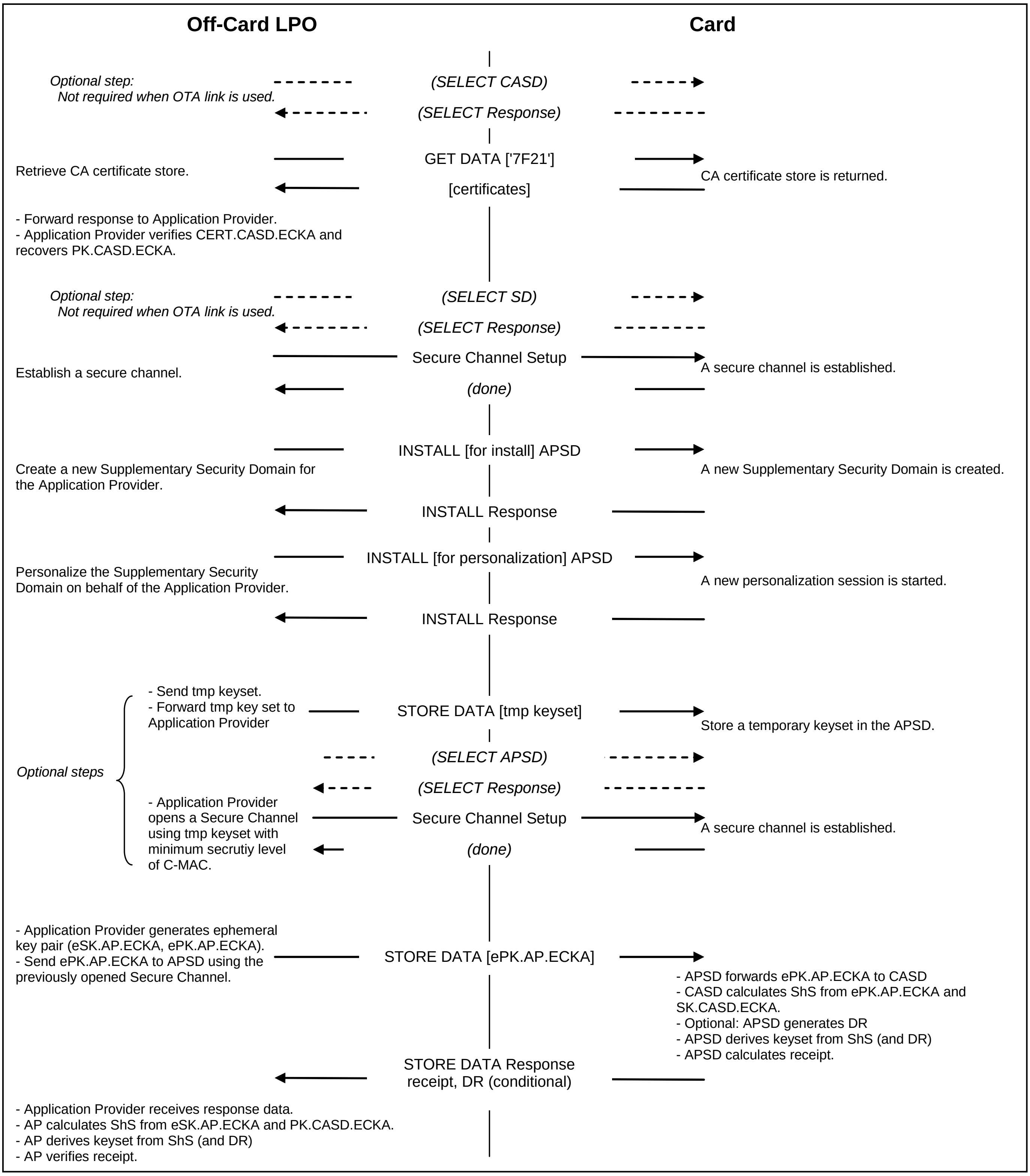
5 Confidential Setup of Secure Channel Keys using ECKA-EG

Enabled by the presence and the capabilities of the CASD, one of the following scenarios may be used to confidentially personalize a set of Secure Channel Keys of a Security Domain using ECKA-EG.

In the following diagram, ECKA-EG is used immediately after creation of a new SD. However, ECKA-EG may be applied at any time during the lifetime of a SD, e.g. to replace a temporary key set that was loaded after SD creation or to renew a key set at a later point in time.

This diagram also shows how a Link Platform Operator (LPO) that may not be trusted by the Application Provider is involved in this scenario.

Figure 5-1: Scenario #3, Using ECKA-EG Scheme


Off-Card LPO
Card
Optional step:
Not required when OTA link is used.
(SELECT CASD)
(SELECT Response)
GET DATA ['7F21']
Retrieve CA certificate store.
[certificates]
CA certificate store is returned.
- Forward response to Application Provider.
- Application Provider verifies CERT.CASD.ECKA and recovers PK.CASD.ECKA.
Optional step:
Not required when OTA link is used.
(SELECT SD)
(SELECT Response)
Secure Channel Setup
Establish a secure channel.
A secure channel is established.
(done)
INSTALL [for install] APSD
Create a new Supplementary Security Domain for the Application Provider.
A new Supplementary Security Domain is created.
INSTALL Response
INSTALL [for personalization] APSD
Personalize the Supplementary Security Domain on behalf of the Application Provider.
A new personalization session is started.
INSTALL Response
- Send tmp keyset.
- Forward tmp key set to Application Provider
STORE DATA [tmp keyset]
Store a temporary keyset in the APSD.
Optional steps
(SELECT APSD)
(SELECT Response)
- Application Provider opens a Secure Channel using tmp keyset with minimum secrutiy level of C-MAC.
Secure Channel Setup
A secure channel is established.
(done)
- Application Provider generates ephemeral key pair (eSK.AP.ECKA, ePK.AP.ECKA).
- Send ePK.AP.ECKA to APSD using the previously opened Secure Channel.
STORE DATA [ePK.AP.ECKA]
- APSD forwards ePK.AP.ECKA to CASD
- CASD calculates ShS from ePK.AP.ECKA and SK.CASD.ECKA.
- Optional: APSD generates DR
- APSD derives keyset from ShS (and DR)
- APSD calculates receipt.
STORE DATA Response receipt, DR (conditional)
- Application Provider receives response data.
- AP calculates ShS from eSK.AP.ECKA and PK.CASD.ECKA.
- AP derives keyset from ShS (and DR)
- AP verifies receipt.

6 Indication of Card and CASD Capabilities

6.1 Card Capability Information

A card may indicate its support of cipher suites in the Card Capability Information as follows:

Table 6-1: Card Capabilities

Tag	Length	Data / Description	Presence
'67'	var	Card Capability Information	Mandatory
'A0'	var	SCP information for first (or only) SCP	Mandatory
'A0'	var	SCP information for additional SCP	Conditional
…		…	
'81'	3	Privileges that can be assigned to an SSD	Conditional
'82'	3	Privileges that can be assigned to any application	Mandatory
'83'	1	LFDBH algorithm	Mandatory
'84'	var	Cipher suites for LFDB encryption	Conditional
'85'	var	Cipher suites supported for tokens	Conditional
'86'	var	Cipher suites supported for receipts	Conditional
'87'	var	Cipher suites supported for DAPs	Conditional
'88'	var	Key Parameter Reference List	Conditional

Structure of SCP information TLV:

Table 6-2: SCP Information

Tag	Length	Data / Description	Presence
'A0'	var	SCP information	
'80'	1	SCP type ('02', '03', '80', '81')	Mandatory
'81'	var	List of supported options for that protocol (e.g. '15 55' for SCP02)	Mandatory
'82'	var	Supported keys (for SCP03 only)	Conditional
'83'	1	Supported TLS cipher suites (for SCP81 only)	Conditional
'84'	1	Maximum length of TLS secret in bytes (unsigned integer) (for SCP81 only)	Conditional

Coding of supported keys for SCP03:

Table 6-3: Supported Keys for SCP03

b8	b7	b6	b5	b4	b3	b2	b1	Description
–	–	–	–	–	–	–	1	AES-128
–	–	–	–	–	–	1	–	AES-192
–	–	–	–	–	1	–	–	AES-256
x	x	x	x	x	–	–	–	RFU (0)

Additional bytes may be appended in the future.

Coding of LFDBH algorithm:

'01'	SHA-1
'02'	SHA-256
'03'	SHA-384
'04'	SHA-512

Coding of cipher suites for LFDB encryption:

Table 6-4: Cipher Suites for LFDB Encryption

b8	b7	b6	b5	b4	b3	b2	b1	Description
–	–	–	–	–	–	–	1	Triple DES with 16 byte key length
–	–	–	–	–	–	1	–	AES-128
–	–	–	–	–	1	–	–	AES-192
–	–	–	–	1	–	–	–	AES-256
1	–	–	–	–	–	–	–	ICV supported for LFDB encryption
–	x	x	x	–	–	–	–	RFU (0)

Additional bytes may be appended in the future.

Coding of cipher suites for signatures used for tokens, receipts and DAPs:

Byte 1:

Table 6-5: Cipher Suites for Signatures – Byte 1

b8	b7	b6	b5	b4	b3	b2	b1	Description
–	–	–	–	–	–	–	1	RSA-1024 / RSASSA-PKCS-v1_5 / SHA-1 (as defined in [GPCS])
–	–	–	–	–	–	1	–	RSA >1024 / RSASSA-PSS / SHA-256 (as defined in this document)
–	–	–	–	–	1	–	–	16 byte key / Single DES plus Final Triple DES MAC (as defined in [GPCS])
–	–	–	–	1	–	–	–	CMAC using AES-128 (as defined in [Amd D])
–	–	–	1	–	–	–	–	CMAC using AES-192 (as defined in [Amd D])
–	–	1	–	–	–	–	–	CMAC using AES-256 (as defined in [Amd D])
–	1	–	–	–	–	–	–	ECDSA using ECC-256 and SHA-256 (as defined in this document)
1	–	–	–	–	–	–	–	ECDSA using ECC-384 and SHA-384 (as defined in this document)

Byte 2:

Table 6-6: Cipher Suites for Signatures – Byte 2

b8	b7	b6	b5	b4	b3	b2	b1	Description
–	–	–	–	–	–	–	1	ECDSA using ECC-512 and SHA-512 (as defined in this document)
–	–	–	–	–	–	1	–	ECDSA using ECC-521 and SHA-512 (as defined in this document)
x	x	x	x	x	x	–	–	RFU (0)

Byte 2 may be missing if its content is zero. Additional bytes may be appended in the future.

Coding of Key Parameter Reference List:

Sequence of all global Key Parameter References for the sets of ECC curve parameters that are available on the card.

6.2 CASD Capability Information

To expose its capabilities, the CASD shall provide off-card entities with CASD Capability Information in the following Security Domain Management Data, upon reception of a GET DATA command for tag '66' (as described in sections H.2 and H.3 of [GPCS]):

Table 6-7: CASD Management Data

Tag	Length	Data / Description	Presence
'66'		Security Domain Data	Mandatory
'73'		SD Recognition Data	Mandatory
'06'	'07'	'2A 8648 86FC6B 01'	Mandatory
'60'	'0B'	SD Management Type & Version	Mandatory
'06'	'09'	'2A 8648 86FC6B 02 02 02'	Mandatory
'63'	'09'	SD Identification Scheme	Mandatory
'06'	'07'	'2A 8648 86FC6B 03'	Mandatory
'64'	'0B'	SD Secure Channel Protocol & Options	Mandatory
…	…	…	…
'65'	'0B'	SD Configuration Details	Mandatory
'06'	'09'	'2A 8648 86FC6B 05' + CASD Capability Information	Mandatory

CASD Capability Information is encoded on 2 bytes as follows:

Table 6-8: CASD Capability Information – Byte 1

b8	b7	b6	b5	b4	b3	b2	b1	Description
–	–	–	–	–	–	–	1	Scenario #1 according to [Amd A]
–	–	–	–	–	–	1	–	Scenario #2.A using RSA1024 according to [Amd A]
–	–	–	–	–	1	–	–	Scenario #2.B using RSA1024 according to [Amd A]
–	–	–	–	1	–	–	–	Scenario #3 using ECC-256 (as defined in this document)
–	–	–	1	–	–	–	–	Scenario #3 using ECC-384 (as defined in this document)
–	–	1	–	–	–	–	–	Scenario #3 using ECC-512 (as defined in this document)
–	1	–	–	–	–	–	–	Scenario #3 using ECC-521 (as defined in this document)
x	–	–	–	–	–	–	–	RFU (0)

Table 6-9: CASD Capability Information – Byte 2

b8	b7	b6	b5	b4	b3	b2	b1	Description
–	–	–	–	–	–	–	x	This bit is only relevant if B1.b1 is set to 1, else it shall be set to 0.
–	–	–	–	–	–	–	0	Scenario #1 using Triple DES with 16 byte key length according to [Amd A]
–	–	–	–	–	–	–	1	Scenario #1 using RSA1024 according to [Amd A]
x	x	x	x	x	x	x	–	RFU (0)

Additional bytes may be appended in the future.

If the card supports RSA cryptography and the CASD supports scenario #1 then the CASD shall support the PK variant of scenario #1 (B2.b1=1).

The CASD shall be personalized so that CASD Capability Information and CASD credentials remain consistent and reflect the scenarios actually supported by the CASD.vz