

DIN EN 419212-2

ICS 35.240.15

Ersatz für
DIN EN 14890-2:2009-03

**Anwendungsschnittstelle für Chip-Karten, die zur Erzeugung
qualifizierter elektronischer Signaturen verwendet werden –
Teil 2: Zusätzliche Dienste;
Englische Fassung EN 419212-2:2014**

Application Interface for smart cards used as Secure Signature Creation Devices –
Part 2: Additional services;
English version EN 419212-2:2014

Interface applicative des cartes à puces utilisées comme dispositifs de création de
signature numérique sécurisés –
Partie 2: Services complémentaires;
Version anglaise EN 419212-2:2014

Gesamtumfang 131 Seiten

DIN-Normenausschuss Informationstechnik und Anwendungen (NIA)

Nationales Vorwort

Dieses Dokument (EN 419212-2:2014) wurde vom Technischen Komitee CEN/TC 224 „Persönliche Identifikation, elektronische Signatur, maschinenlesbare Karten sowie zugehörige Geräteschnittstellen und Verfahren“ erarbeitet, dessen Sekretariat vom AFNOR (Frankreich) gehalten wird.

Das zuständige deutsche Normungsgremium ist der Arbeitsausschuss NA 043-01-17-04 UA „Austauschprotokolle bei Chipkarten“ im DIN-Normenausschuss Informationstechnik und Anwendungen (NIA).

Dieses Dokument ersetzt DIN EN 14890-2:2009-03.

DIN EN 419212-2 ist Teil einer zweiteiligen Normenreihe zur Festlegung von Anwendungsschnittstellen für Chipkarten, die zur Erzeugung qualifizierter elektronischer Signaturen verwendet werden.

DIN EN 419212-1 soll die Implementierung des europäischen rechtlichen Rahmens für elektronische Signaturen unterstützen, indem Sie die funktions- und sicherheitsbezogenen Anforderungen an eine Chipkarte zur Verwendung als sichere Signaturerstellungseinheit (Secure Signature Creation Device, SSCD) entsprechend den Bedingungen der Richtlinie 1999/93/EG des Europäischen Parlaments und des Rates vom 13. Dezember 1999 über gemeinschaftliche Rahmenbedingungen für elektronische Signaturen definiert. Eine mit der Norm übereinstimmende Karte wird in der Lage sein, eine „qualifizierte elektronische Signatur“ (QES) zu erzeugen, die Artikel 5.1 der Signaturrechtlinie erfüllt und daher als zu handschriftlichen Signaturen äquivalent betrachtet werden kann.

DIN EN 419212-2 legt Mechanismen zur Unterstützung weiterer Dienste zur generischen Identifizierung, Authentisierung, Vertraulichkeit, Signaturüberprüfung und zum Datenschutz fest.

DIN EN 419212 definiert eine Menge von Diensten, die die Entwicklung interoperabler Karten, die von einem beliebigen Kartensektor ausgegeben werden, ermöglicht. Sie beschreibt eine Anwendungsschnittstelle und das Verhalten von sicheren Signaturerstellungseinheiten, d.h. eine Implementierung in systemspezifische und interpreterbasierten Karten sollte somit möglich sein.

Das Präsidium des DIN hat mit seinem Beschluss 1/2004 festgelegt, dass von dem in den Regeln der europäischen Normungsarbeit von CEN/CENELEC verankerten Grundsatz, wonach Europäische Normen in den drei offiziellen Sprachen Deutsch, Englisch, Französisch veröffentlicht werden, in begründeten Ausnahmefällen abgewichen und auf die deutsche Sprachfassung verzichtet werden kann. Daher wurden von dieser Norm nur die Begriffe übersetzt (siehe nationaler Anhang NA).

Änderungen

Gegenüber DIN EN 14890-2:2009-03 wurden folgende Änderungen vorgenommen:

- a) es wurden folgende Dienste zum Datenschutz hinzugefügt:
 - 1) Dienste zur Anonymisierung und Pseudonymität;
 - 2) Zusatzdatenübertragung z. B. zur Altersüberprüfung;
 - 3) e-Services mit vertrauenswürdigen Dritten;
 - 4) e-Service mit 2 Beteiligten

Frühere Ausgaben

DIN EN 14890-2: 2009-03

Nationaler Anhang NA (informativ)

Begriffe

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

ANMERKUNG Die Definitionen entsprechen denen in der Überarbeitung von ISO/IEC 7816-4 angegebenen.

3.1

Anonymität

Zusicherung, dass ein Benutzer ein Betriebsmittel oder einen Dienst ohne Offenlegung seiner Benutzeridentität nutzen kann

3.2

Anonymisierung

Prozess, der die Zuordnung zwischen einer personenbezogenen Datenmenge und einem Datensubjekt entfernt

3.3

anonymisierte Daten

ehemals mit einer Person verknüpfte Daten, die jetzt nicht mehr mit dieser Person in Verbindung gebracht werden können

3.4

anonyme Daten

Daten, die nicht mit einer bestimmten Person verknüpft werden können

3.5

externe C/S-Authentisierung

Authentisierung des Servers durch den Client

Anmerkung 1 zum Begriff: Der Client gilt als eine Kombination aus dem PC und der Chipkarte. Diese externe Authentisierung liegt außerhalb des Anwendungsbereichs dieser Norm.

3.6

interne C/S-Authentisierung

Authentisierung des Clients durch den Server

Anmerkung 1 zum Begriff: Der Client gilt als eine Kombination aus dem PC und der Chipkarte.

3.7

Folgenlosigkeit

Sicherheitseigenschaft eines Protokolls, die zusichert, dass die Offenlegung eines privaten Langzeitschlüssels es einem Angreifer nicht ermöglicht, die Sicherheitseigenschaften für die Ausführung des Protokolls, die in der Vergangenheit festgelegt wurden, zum Beispiel durch Neuberechnung früherer abgeleiteter Schlüssel, zu gefährden

3.8

Identifizierung

eindeutige Zuordnung einer Menge von beschreibenden Parametern zu einer Person innerhalb eines vorgegebenen Kontextes

3.9

Schnittstellengerät

FD

Gerät oder Instanz, das/die zur externen Welt (außerhalb der Chipkarte) gehört

3.10

Datenschutz

Anspruch von Personen, Gruppen oder Organisationen, selbst zu bestimmen, wann, wie und in welchem Umfang Informationen über sie anderen mitgeteilt werden

3.11

Pseudonymität

Zusicherung, dass ein Benutzer ein Betriebsmittel oder einen Dienst ohne Offenlegung seiner Benutzeridentität nutzen kann

3.12

gesicherter Kanal

Kommunikationsverbindung zwischen der Chipkarte und einem Sicherheitsmodul (möglicherweise ebenfalls eine Chipkarte), die Authentizität und/oder Integrität und/oder Geheimhaltung bereitstellt

3.13

Nichtverkettbarkeit

Zusicherung, dass ein Benutzer eine Ressource oder einen Dienst mehrere Male nutzen kann, ohne dass andere diese Nutzungen miteinander verknüpfen können [14]

3.14

Gebrauch von Wendungen in der vorliegenden Norm

Die Schlüsselworte „ERFORDERLICH“ (en: REQUIRED“), „MUSS“ (en: SHALL), „DARF NICHT“ (en: SHALL NOT), „SOLLTE“ (en: SHOULD), „SOLLTE NICHT“ (en: SHOULD NOT), „EMPFOHLEN“ (en: RECOMMENDED), „DARF“ (en: MAY) und „WAHLWEISE“ (en: OPTIONAL) im vorliegenden Dokument sind nach der Beschreibung in [21] zu interpretieren

Anmerkung 1 zum Begriff: Siehe die folgende Liste von Schlüsselwörtern.

MUSS:	Dieses Wort oder die Benennung „ERFORDERLICH“ bedeutet, dass die Definition eine absolute Anforderung der Spezifikation darstellt.
DARF NICHT:	Dieser Ausdruck bedeutet, dass die Definition ein absolutes Verbot der Spezifikation darstellt.
SOLLTE:	Dieses Wort oder das Adjektiv „EMPFOHLEN“ bedeutet, dass unter bestimmten Umständen berechnete Gründe dafür bestehen, ein bestimmtes Element zu vernachlässigen, aber die vollständigen Auswirkungen nachvollzogen und sorgfältig abgewogen werden müssen, bevor ein anderer Ablauf gewählt wird.
SOLLTE NICHT:	Dieser Ausdruck oder das Adjektiv „NICHT EMPFOHLEN“ bedeutet, dass unter bestimmten Umständen berechnete Gründe dafür bestehen, dass das jeweilige Verhalten zulässig oder sogar zweckdienlich ist, aber die vollständigen Auswirkungen sollten nachvollzogen und der Fall sorgfältig abgewogen werden, bevor ein mit diesem Kennzeichen beschriebenes Verhalten implementiert wird.
DARF:	Dieses Wort oder das Adjektiv „WAHLWEISE“ bedeutet, dass ein Element wahlweise benutzt werden kann. Ein Hersteller kann sich für die Aufnahme des Elements entscheiden, weil es von einem bestimmten Marktsegment gefordert wird oder der Hersteller meint, dass es sein Produkt verbessert, während ein anderer Hersteller das gleiche Element weglassen kann. Eine Implementierung, die eine bestimmte Wahlmöglichkeit nicht enthält, muss so vorbereitet sein, dass sie mit einer anderen Implementierung zusammenarbeiten kann, die die Wahlmöglichkeit enthält, wenn auch unter Umständen mit einer verringerten Funktionalität.

[...]

Eckige Klammern zeigen eine Wahlfreiheit an. Das kann entweder sein:

- eine *Auswahl von Werten* in einem vorgegebenen Bereich [,00' .. ,03'] — typisch für Felder mit fester Länge,
- oder eine Auswahl, ob ein Wert angegeben wird oder nicht — ... || ,80' L80 AlgID || [,83' L83 KeyID] ..., wobei der Datensatz in eckigen Klammern den *wahlweisen* Teil (= im Ermessen der Implementierung) angibt oder *bedingt* ist (= abhängig von einer in der Norm angegebenen Bedingung).

BEDINGUNG

Die Anwendung einer Ausprägung hängt von einer oder mehrerer Bedingungen ab und wird nur vorgenommen, wenn die Bedingung(en) erfüllt werden.

Die zugehörige Bedingung(en) wird immer beschrieben, wenn ein bedingtes Attribut durch eine Ausprägung erstellt wurde. Die Ausprägung darf, abhängig vom Kontext in dem sie erstellt wurde, normativ oder informativ sein, wenn die Bedingungen erfüllt sind.

— Leerseite —

English Version

Application Interface for smart cards used as Secure Signature Creation Devices - Part 2: Additional services

Interface applicative des cartes à puces utilisées comme
dispositifs de création de signature numérique sécurisés -
Partie 2 : Services complémentaires

Anwendungsschnittstelle für Chip-Karten, die zur
Erzeugung qualifizierter elektronischer Signaturen
verwendet werden - Teil 2: Zusätzliche Dienste

This European Standard was approved by CEN on 27 September 2014.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Contents

Foreword.....	5
1 Scope	7
2 Normative references	7
3 Terms and definitions	7
4 Abbreviations and notation	9
5 Additional Service Selection	11
6 Client/Server Authentication	14
6.1 Client/Server protocols	14
6.2 Steps preceding the client/server authentication	15
6.3 Padding format.....	15
6.3.1 PKCS #1 v 1-5 Padding	15
6.3.2 PKCS #1 V 2.x (PSS) Padding.....	16
6.3.3 Building the DSI on ECDSA	17
6.4 Client/Server protocol	18
6.4.1 Step 1 — Read certificate.....	18
6.4.2 Step 2 — Set signing key for client/server internal authentication	19
6.4.3 Step 3 — Internal authentication.....	20
6.4.4 Client/Server authentication execution flow	22
6.4.5 Command data field for the client server authentication	24
7 Role Authentication	25
7.1 Role Authentication of the card	25
7.2 Role Authentication of the server	25
7.3 Symmetrical external authentication	25
7.3.1 Protocol	25
7.3.2 Description of the cryptographic mechanisms	30
7.3.3 Role description.....	30
7.4 Asymmetric external authentication.....	31
7.4.1 Protocol based on RSA.....	31
7.4.2 Protocol based on modular Enhanced Role Authentication (mERA)	34
8 Symmetric key transmission between a remote server and the ICC	49
8.1 Steps preceding the key transport.....	49
8.2 Key encryption with RSA	49
8.2.1 PKCS#1 v1.5 padding.....	50
8.2.2 OAEP padding.....	50
8.2.3 Execution flow.....	51
8.3 Diffie-Hellman key exchange for key encipherment	54
8.3.1 Execution flow.....	56
9 Signature verification	58
9.1 Signature verification execution flow	58
9.1.1 Step 1: Receive Hash	59
9.1.2 Step 2: Select verification key	60
9.1.3 Step 3: Verify digital signature	61
10 Certificates for additional services	62
10.1 File structure	63
10.2 EF.C_X509.CH.DS	63
10.3 EF.C.CH.AUT	63
10.4 EF.C.CH.KE.....	63
10.5 Reading Certificates and the public key of CAs.....	64
11 Privacy Context functions	65

11.1	Introduction.....	65
11.2	Auxiliary Data Comparison.....	65
11.2.1	Presentation of the auxiliary data.....	66
11.2.2	Age Verification	68
11.2.3	Document Validation.....	69
11.3	Restricted Identification.....	70
11.3.1	Command APDU for Step RI:1	73
11.3.2	Command APDU for Step RI:2	74
11.4	eServices with trusted third party protocol.....	77
11.4.1	mERA-based eServices with trusted third party protocol	78
11.4.2	mEAC-based eServices with trusted third party.....	83
11.5	eServices with two party protocols	86
11.5.1	mEAC-based eServices with on-line two party protocol.....	86
11.5.2	mEAC-based eServices with off-line two party protocol	87
12	APDU data structures	89
12.1	Algorithm Identifiers	89
12.2	CRTs	89
12.2.1	CRT DST for selection of ICC's private client/server auth. key	89
12.2.2	CRT AT for selection of ICC's private client/server auth. key	89
12.2.3	CRT CT for selection of ICC's private key	90
12.2.4	CRT DST for selection of IFD's public key (signature verification)	90
Annex A	(normative) Security Service Descriptor Templates.....	91
A.1	Security Service Descriptor Concept	91
A.2	SSD Data Objects	92
A.2.1	DO Extended Header List, tag '4D'	92
A.2.2	DO Instruction set mapping (ISM), tag '80'	92
A.2.3	DO Command to perform (CTP), tag '52' (refer to ISO/IEC 7816-6).....	92
A.2.4	DO Algorithm object identifier (OID), tag '06' (refer to ISO/IEC 7816-6)	92
A.2.5	DO Algorithm reference, tag '81'	92
A.2.6	DO Key reference, tag '82'	93
A.2.7	DO FID key file, tag '83'	93
A.2.8	DO Key group, tag '84'	93
A.2.9	DO FID base certificate file, tag '85'	93
A.2.10	DO FID adjoined certificate file, tag '86'	93
A.2.11	DO Certificate reference, tag '87'	93
A.2.12	DO Certificate qualifier, tag '88'	93
A.2.13	DO FID for file with public key of the certification authority PK(CA), tag '89'	93
A.2.14	DO PIN usage policy, tag '5F2F'	93
A.2.15	DO PIN reference, tag '8A'	94
A.2.16	DO Application identifier (AID), tag '4F' (refer to ISO/IEC 7816-6).....	94
A.2.17	DO CLA coding, tag '8B'	94
A.2.18	DO Status information (SW1-SW2), tag '42' (refer to ISO/IEC 7816-6)	94
A.2.19	DO Discretionary data, tag '53' (refer to ISO/IEC 7816-6).....	94
A.2.20	DO SE number, tag '8C'	94
A.2.21	DO SSD profile identifier, tag '8D'	95
A.2.22	DO FID mapping, tag '8E'	95
A.3	Location of the SSD templates	95
A.4	Examples for SSD templates.....	95
Annex B	(informative) Security environments	97
B.1	Definition of CRTs (examples)	98
B.1.1	CRT for Authentication (AT)	99
B.1.2	CRT for Cryptographic Checksum (CCT).....	100
B.1.3	CRT for Digital Signature (DST)	101
B.1.4	CRT for confidentiality (CT).....	102
B.2	Security Environments (example)	103
B.2.1	Security Environment #10	103
B.2.2	Security Environment #11	104
B.3	Coding of access conditions (example).....	104

B.3.1 Access Conditions..... 105

B.3.2 Access rule references 106

B.3.3 Access conditions for EF.ARR..... 107

B.3.4 EF.ARR records 107

Annex C (normative) Algorithm Identifiers — Coding and specification 110

Annex D (informative) Example of DF.CIA..... 117

Annex E (informative) Build scheme for object identifiers defined by EN 14890 122

Bibliography 124

Foreword

This document (EN 419212-2:2014) has been prepared by Technical Committee CEN/TC 224 "Personal identification, electronic signature and cards and their related systems and operations", the secretariat of which is held by AFNOR.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by June 2015 and conflicting national standards shall be withdrawn at the latest by June 2015.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN [and/or CENELEC] shall not be held responsible for identifying any or all such patent rights.

This document supersedes EN 14890-2:2008.

This document has been prepared under a mandate given to CEN by the European Commission and the European Free Trade Association.

EN 419212, *Application Interface for smart cards used as Secure Signature Creation Devices*, consists of two parts:

- *Part 1: Basic services* which describes the specifications for IAS based services on smart cards to be used in compliance to the requirements of Article 5.1 of the Electronic Signature Directive; and
- *Part 2: Additional services* [the present document] which describes other services that may be used in conjunction with all, some or none of the services described in Part 1.

This standard supports services in the context of IAS Identification, **A**uthentication and Electronic **S**ignature (IAS) services, as well as other services.

In EN 419212-1, the standard allows to support the implementation of the European legal framework for electronic signatures, defining the functional and security features for a smart card intended to be used as a Secure Signature Creation Device according to the Terms of the European Directive on Electronic Signature 1999/93/EC. A card compliant to the standard will be able to produce a "Qualified Electronic Signature (QES)" that fulfils the requirements of Article 5.1 of the Electronic Signature Directive and therefore can be considered equivalent to hand-written signatures.

In EN 419212-2, the standard specifies mechanisms to support other services like generic Identification, Authentication, confidentiality, signature verification services and privacy features.

EN 419212 defines a set of services that will enable the development of interoperable cards issued by any card industry sector. The standard will describe an application interface and behavior of the SSCD, i.e. it should be possible to implement it on native and interpreter based cards.

Compared with the 2008 versions of EN 14890, the following broad change has been made:

The scope of the standard was enhanced through new mechanisms in the field of password based mechanisms and privacy.

Regarding EN 419212-1, the most significant technical changes that have been made are the following ones:

- new algorithms added to device authentication protocols (e.g. AES, ELC);
- added AES to secure messaging;

- introduced password based mechanisms (PACEv2);
- updating references to their latest releases;
- algorithm Identifier coding;
- recommendation for making best use of device authentication protocols.

Regarding EN 419212-2, the most significant technical changes that have been made are the following ones:

- a) Added privacy services including:
 - 1) anonymity and pseudonymity services;
 - 2) auxiliary data transmission e.g. for Age verification;
 - 3) e-Services with trusted third party;
 - 4) e-Services with 2-parties.

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

1 Scope

This European Standard contains Identification, Authentication and Digital Signature (IAS) services in addition to the SSCD mechanisms already described in EN 419212-1 to enable interoperability and usage for IAS services on a national or European level.

It also specifies additional mechanisms like key decipherment, Client Server authentication, identity management and privacy related services.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

EN 419212-1:2014, *Application Interface for smart cards used as Secure Signature Creation Devices — Part 1: Basic services*

ISO/IEC 7816-4:2013, *Identification cards — Integrated circuit(s) cards with contacts — Part 4: Organization, security and commands for interchange*

ISO/IEC 7816-6:2006, *Identification cards — Integrated circuit(s) cards with contacts — Part 6: Interindustry data elements for interchange*

ISO/IEC 7816-8:2004, *Integrated circuit(s) cards with contacts — Part 8: Commands for security operations*

ISO/IEC 9796 (all parts), *Information technology — Security techniques — Digital signature schemes giving message recovery*

ISO/IEC 9797-1, *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

NOTE These definitions are in compliance with those given in the revision of ISO/IEC 7816-4.

3.1

anonymity

assurance in which a user may use a resource or service without disclosing the user's identity

3.2

anonymization

process that removes the association between an identifying data set and a data subject

3.3

anonymized data

data that was once linked to an individual but can now no longer be related to them

3.4

anonymous data

data that cannot be linked to a specific individual

3.5

C/S external authentication

authentication of the server by the client

Note 1 to entry: The client is regarded as the combination of the PC and the ICC. This external authentication is out of the scope of this specification.

3.6

C/S internal authentication

authentication of the client by the server

Note 1 to entry: The client is regarded as the combination of the PC and the ICC.

3.7

forward secrecy

security property of a protocol, that guarantees that the disclosure of long-term private key does not enable an opponent to compromise the secrecy property of the executions of the protocol made in the past, for example, by re-computing previously derived keys

3.8

identification

unique association of a set of descriptive parameters to an individual within a given context

3.9

IFD

device or entity that belongs to the external world (outside the ICC)

3.10

privacy

claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others

3.11

pseudonymity

ensurance that a user may use a resource or service without disclosing its user identity

3.12

secured channel

communication link between the ICC and a security module (possibly also an ICC) that provides authenticity and/or integrity and/or confidentiality

3.13

unlinkability

assurance that a user may make multiple uses of resources or services without others being able to link these uses together [14]

3.14

usage of expressions in this standard

use of the key words "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document as to be interpreted as described in [21]

Note 1 to entry: See the following list of key-words:

SHALL:	This word, or the terms " REQUIRED " or " SHALL ", means that the definition is an absolute requirement of the specification.
SHALL NOT:	This phrase, or the phrase " SHALL NOT ", means that the definition is an absolute prohibition of the specification.
SHOULD:	This word, or the adjective " RECOMMENDED ", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications will be understood and carefully weighed before choosing a different course.
SHOULD NOT:	This phrase, or the phrase " NOT RECOMMENDED " means that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full

implications should be understood and the case carefully weighed **before implementing any behaviour described with this label.**

MAY: This word, or the adjective "**OPTIONAL**", means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item.

An implementation which does not include a particular option will be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality.

[...] Square brackets indicate a freedom of choice. This can be either a:
- *choice of values* in a given range ['00' .. '03'] – typical for fixed length fields,
- or a choice whether to present a value or not – ... || '80' L80 AlgID || ['83' L83 KeyID] whereas the set of data in square brackets is the *optional* part (= at the discretion of the implementation) or *conditional* (= depending on a condition given in the standard).

CONDITIONAL: The application of a specification depends on one or more conditions and shall only be applied if the condition(s) are met.

The associated condition(s) are always described if a *conditional* attribute is made with a specification. If the conditions are met, the specification may be *normative* or *informative* depending on the context in which the specification is made.

4 Abbreviations and notation

For the purposes of this document, the following symbols and abbreviations apply.

APDU	Application Protocol Data Unit
ARR	Access Rule Record
AT	Authentication Template
C/S	Client-Server
CA	Certification Authority
CC	Cryptographic Checksum
CCT	Cryptographic Checksum Template
CIA	Cryptographic Information Application
CLA	Class byte
CMS	Card Management System
CRT	Control Reference Template
CT	Confidentiality Template
CV	Card Verifiable
D[key](msg)	Decipherment of <msg> with <key>
DF	Dedicated File
DH	Diffie-Hellman
DO	Data Object
DS[key](msg)	Digital Signature of <msg> with <key>
DSI	Digital Signature Input
DST	Digital Signature Template
E[key](msg)	Encipherment of <msg> with <key>
ECDSA	Signature Scheme based on elliptic curve cryptography (ELC)
EF	Elementary File

FCI	File Control Information
FCP	File Control Parameters
H_<id>	Hash function using <id> algorithm
ICC	Integrated Circuit(s) Card
ID	Identifier
IFD	Interface Device
INS	Instruction byte
KE	Key Encipherment
KEI	Key Encipherment Input Format
KID	Key Identifier
MD5	Message Digest 5 (hash algorithm)
MF	Master File
OAEP	Optimal Asymmetric Encryption Padding
P1-P2	Parameter bytes
PI	Padding Indicator
PrK	Private Key
PuK	Public Key
PKI	Public Key Infrastructure
PKCS	Public Key Cryptography Standards
PBM	Password based mechanism
PSO	PERFORM SECURITY OPERATION
PSS	Probabilistic Signature Scheme
RA	Role authentication
RCA	Root CA
RFU	Reserved for Future Use
RND	Random Number
RSA	Cryptographic algorithm invented by Ronald Rivest, Adi Shamir and Leonard Adleman
SE	Security Environment
SFI	Short File Identifier
SHA	Secure Hash Algorithm
SK	Secret Key
SN	Serial Number
SM	Secure Messaging
SSCD	Secure Signature Creation Device
SSD	Security Service Descriptor
SW1-SW2	Status bytes
TDES	Triple-DES, this standard only considers the 2-key variant
UQB	Usage Qualifier Byte

5 Additional Service Selection

Additional services are typically used in the context of applications that use digital signatures.

A well known additional service is the **client/server authentication**. In this case, the ICC is used as a crypto toolbox, e.g. in order to encrypt a challenge with a private key, being stored in the ICC. This is particularly helpful in applications, where a tamper resistant device is required for client/server authentication. A secure ICC has the necessary tamper resistant quality and may therefore be used efficiently to support the application in this context.

Document decryption is another known service which may be performed by the IFD. A terminal application receives a document, typically encrypted with a symmetric key. The symmetric key is also provided encrypted with a public key. The ICC contains the appropriate private key, deciphers the symmetric key and returns it to the terminal application.

While the typical usage of a signature card is the generation of a digital signature, an application might want to verify a signature with a public key, being stored in the ICC. In this case an additional service is invoked for **signature verification**.

ICCs used as national identification cards, travel documents or driving licences generally provide additional applications to enable **eServices** (e.g. eGovernment, eBusiness,...) including an ESIGN application. In the eID card context new privacy issues are to be put into account, e.g. user tracking, data minimizing, unlinkability of transactions or domain specific identifiers. This standard specifies privacy preserving protocols and mechanisms as additional services.

Additional services provided in the ICC mandate the existence of an appropriate security environment. Associated security environments are described in Annex B "(informative) Security environments".

In addition to the descriptive information found in DF.CIA (refer to Clause 16 "Cryptographic Information Application" in EN 419212-1:2014) information might be required that can be presented in Security Service Descriptors. The concept of Security Service Descriptors is described in Annex A.

A user verification may be required prior to the usage of additional services. The password for this user verification shall be different from the password used for the signature generation. This is to maintain the purpose of the signature generation password for the sole purpose of a 'declaration of will' in the case of a signature generation.

Figure 1 shows an execution flow for an additional service. The corresponding technical implementation is given in this document.

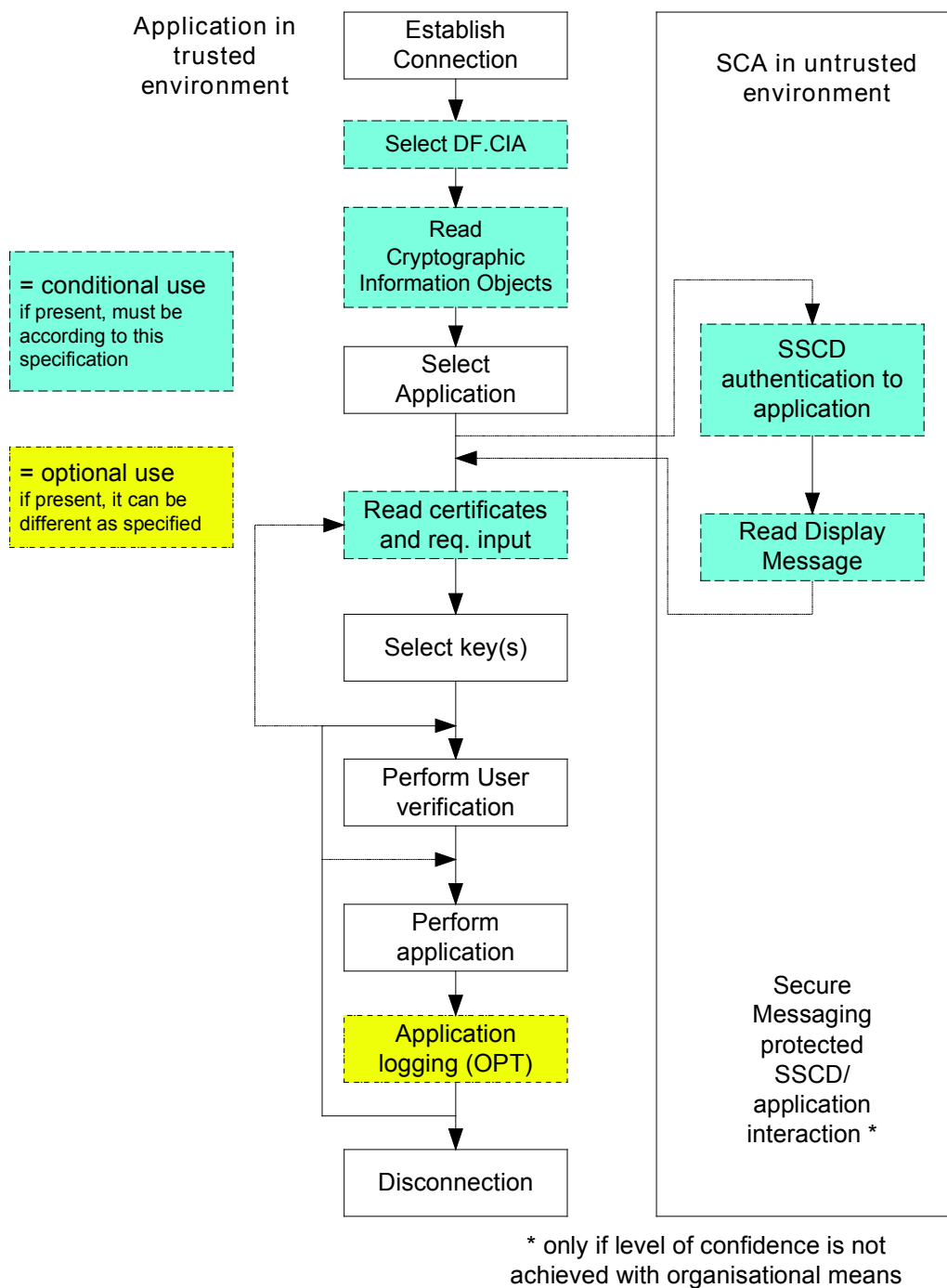


Figure 1 — Interaction sequences between application and SSCD

As the standard specifies various mechanisms for device and user authentication with a number of resulting combinations, Figure 2 shows execution flows for typical signature cards in different security and privacy context.

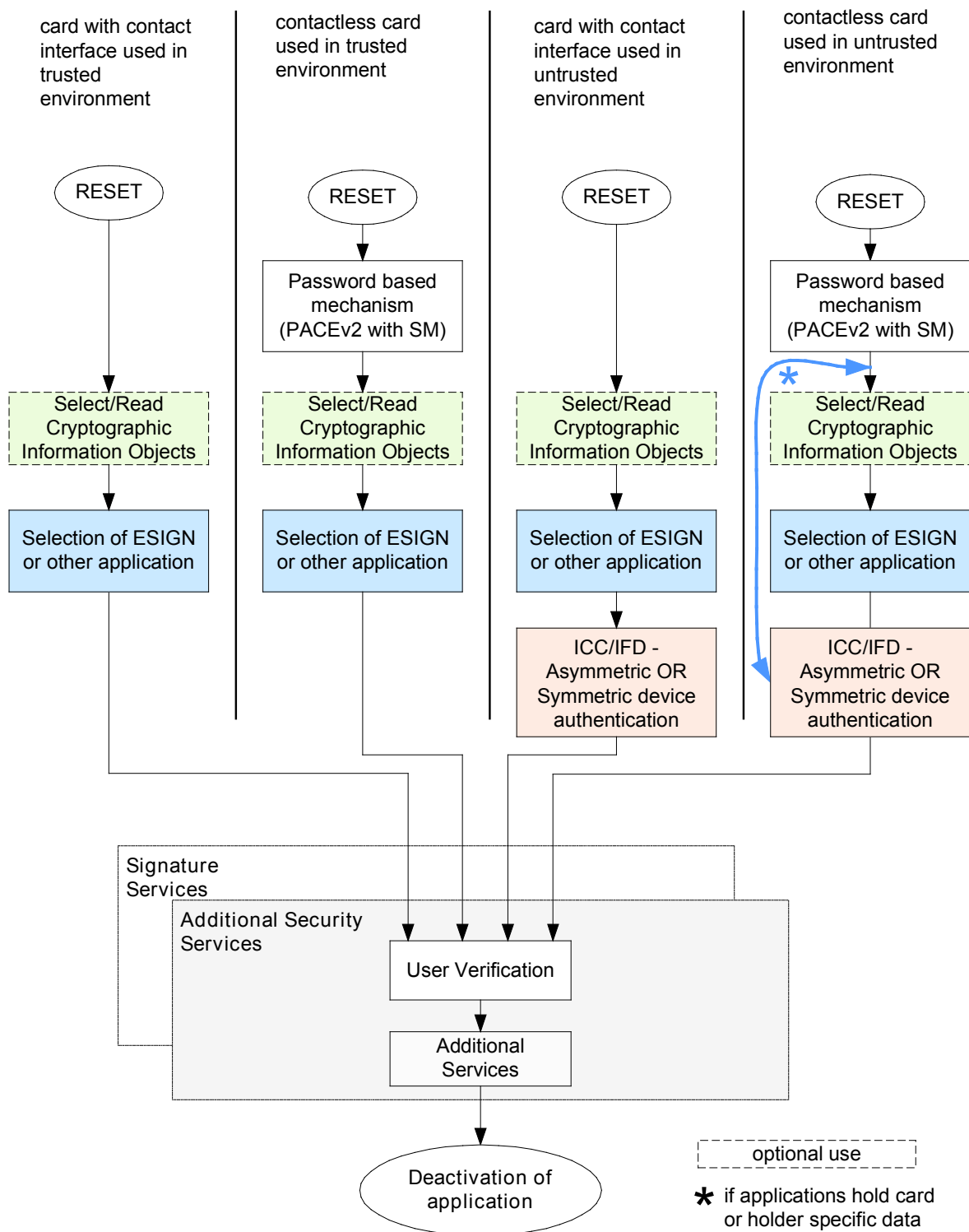


Figure 2 — Example of additional service selection

Figure 2 shows the selection of additional services in the context of the ESIGN application. User verification might be required for some of the additional services. The detailed access conditions are described in the appropriate security environments.

For security reasons the cryptographic information objects shall not reveal any information which could be used to associated the ICC to the password. If this cannot be guaranteed, the device authentication shall be done prior to reading the cryptographic information objects to avoid that an unauthorized IFD may later reuse that association for further attacks. Alternatively other measures shall be taken to avoid such an attack.

6 Client/Server Authentication

For proving access rights to components such as servers, a PK based authentication procedure has to be performed. Such client/server Authentication (refer to “C/S internal authentication”) is a process, independent from the requirement of device authentication.

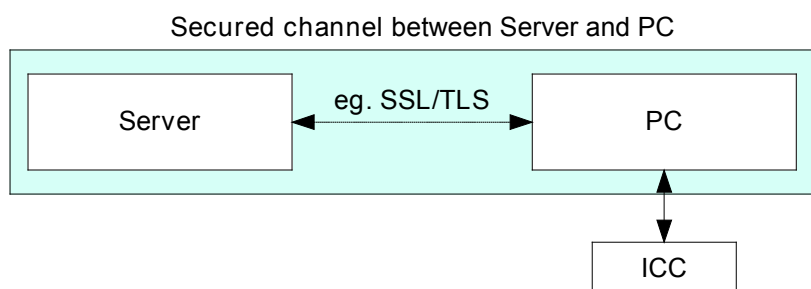


Figure 3 — Example of client/server authentication

In the above example client/server authentication establishes a secured channel between a remote server and a PC. The ICC will be used as a cryptographic toolbox in order to provide the cryptographic functionality to the PC.

This specification does not support the authentication of the server (“C/S external authentication”). The server’s certificate as well as the server protocol is application specific and therefore out of the scope of this document.

6.1 Client/Server protocols

This specification only covers the case, where the ICC performs a digital signature computation on the authentication input contained in the data field of an INTERNAL AUTHENTICATE (COMPUTE DIGITAL SIGNATURE) command. The input is formatted before the private key for authentication is used to form the signature.

The key pair used for client/server authentication shall be different from the device authentication keys and signature generation keys respectively. The public part of this key pair, stored with the distinguished name of the cardholder, is certified by a certificate (typically X.509 [9]). Such a certificate is not interpreted by the ICC.

Relevant authentication procedures are e.g.:

- the PK Kerberos protocol (for logon authentication)
- the SSL/TLS protocol
- the SSH protocol

All the above protocols are based on the same cryptographic algorithms. In particular they all use PKCS #1 padding format in the case of RSA. This specification describes the PKCS #1 padding and C/S authentication based on ECDSA.

6.2 Steps preceding the client/server authentication

The steps preceding a client/server authentication are application specific. Hence this specification does not mandate the existence of those steps.

The access conditions proposed in Annex B “(informative) Security environments” specify a user verification as a mandatory step prior to client/server authentication.

The reference to the password to be used for user verification in the context of client/server authentication is described in the information of DF.CIA.

6.3 Padding format

6.3.1 PKCS #1 v 1-5 Padding

The DSI generation according to PKCS #1 v1.5 uses the padding scheme defined for the EMSA-PKCS-v1_5 encoding defined in [2], Section 9.2 (starting with step 3 of the encoding method). The padding is applied directly to the Authentication Input T.

Figure 4 shows an example for DSI generation according to PKCS #1 V1.5. In case of RSA, the authentication input T is formatted according to [2] PKCS #1, Version 2.1, Chapter 9.2 “Emsa-PKCS1-v1-5”. For particular Authentication input T refer to 6.4.5 “Command data field for the client server authentication”.

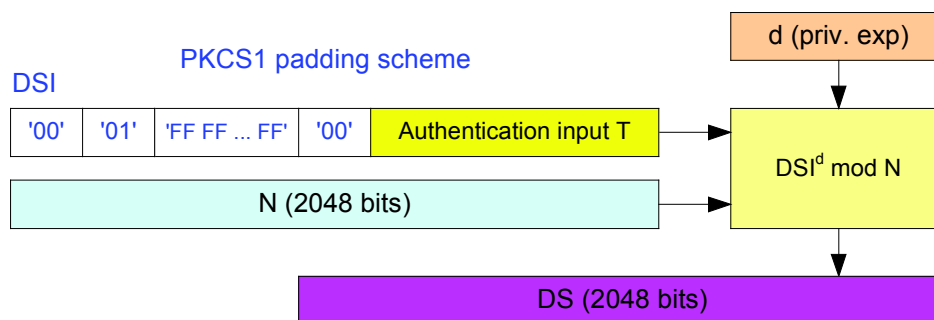


Figure 4 — Example for 2048 bit DSI according to PKCS #1 V1.5

The padding is realized through an octet string consisting of octets with value 'FF' (length ≥ 8). Due to security reasons the authentication input shall be smaller or equal to 33 % of the length of the modulus. The formatted octet string shall consist of k octets where k is the length in octets of the modulus of the private key for authentication.

The digest info is described in 14.3.3 “Digest Info for SHA-X” in EN 419212-1:2014.

6.3.2 PKCS #1 V 2.x (PSS) Padding

The DSI generation according to PKCS1-PSS uses the padding scheme defined for the EMSA-PSS encoding defined in [2] section 9.1. There are two modes of operation, each indicated by a separate set of algorithm IDs see Table C.17 – 1-byte Algorithm-ID coding.

The DSI format according to PKCS #1 V 2.1 has the following structure. The message M represents the authentication input T.

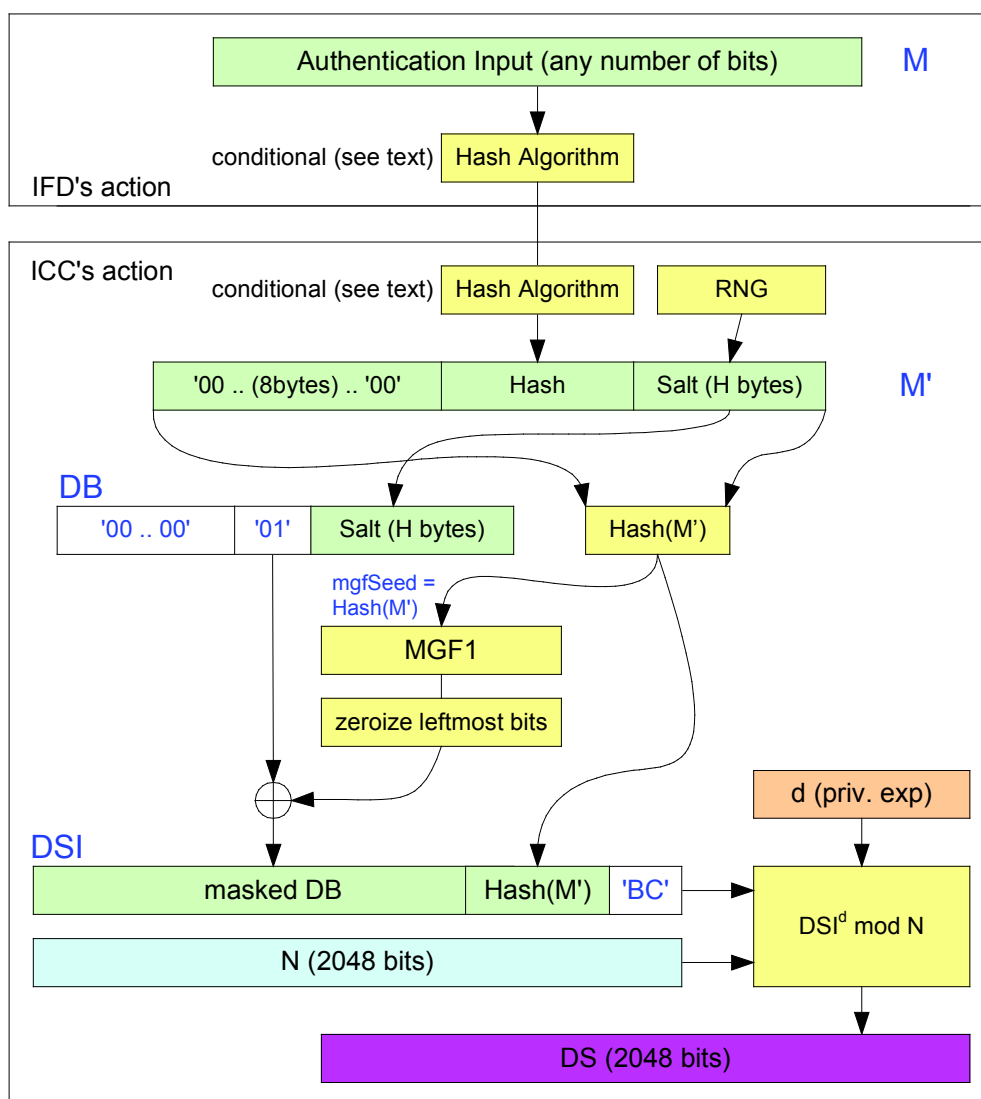


Figure 5 — Example for 2048 bit DSI according to PKCS #1 V2.1

Figure 5 shows an example for the DSI computation according to PKCS #1 V2.1. Building the first hash in the IFD is an optional step and appropriate, if the message M is large and the transmission of this large message M would required exhaustive data transmission to the ICC.

The hashing function used in MGF1 is the same as the one used to hash the authentication input. The $[8 \times \text{Key.ModulusByteLength} - (\text{Key.ModulusBitLength} - 1)]$ leftmost bits of the output of the MGF1 function are set to zero to provide a DSI input being arithmetically smaller than the modulus N. The MGF1 function is described in [2] [PKCS 1 — V.2.1, Chapter B.2.1].

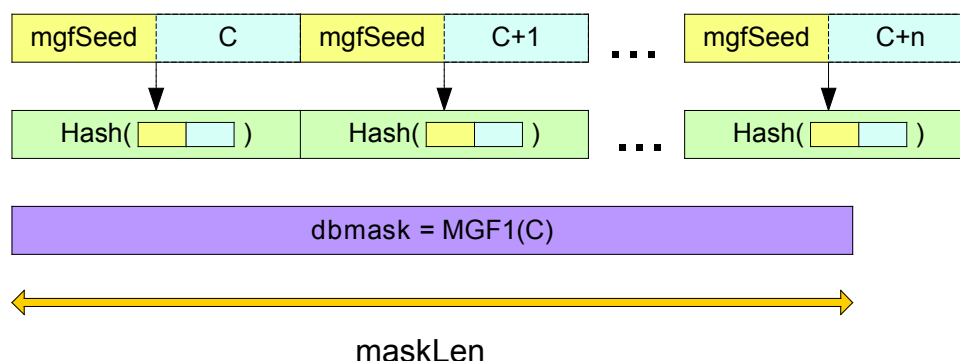


Figure 6 — Example for the mask generating function MGF1

Figure 6 shows an example for the mask generating function MGF1. The length of the salt is identical to the Digest Length H of the hash algorithm. The length of DB computes from

$$\text{Length}(DB) = N - H - 1 = \text{maskLen}.$$

where N is the byte length of the modulus and H is the digest length of the hash algorithm.

The length of the *mgfSeed* is identical to H , the length of C is 4 bytes as specified [2]. The initial value of C is zero. The concatenation of [*mgfSeed* || C] pairs is right truncated at the length of *maskLen* to build *dbmask*.

The $[8 \times \text{Key.ModulusByteLength} - (\text{Key.ModulusBitLength} - 1)]$ leftmost bits of *dbmask* are set to zero to provide a DSI input being arithmetically smaller than the modulus N .

The hashing function used in MGF1 is the same as the one used to hash the authentication input. The $[8 \times \text{Key.ModulusByteLength} - (\text{Key.ModulusBitLength} - 1)]$ leftmost bits of the output of the MGF1 function are set to zero to provide a DSI input being arithmetically smaller than the modulus N .

Table 1 — Digital Signature Input (DSI) — Format acc. to PKCS #1 V 2.x

T	L	V
—	—	masked DB = $DB \oplus \text{MGF1}(\text{Hash}(M'), \text{Key.ByteLength} - H - 1)$ Hash(M') 'BC' = Padding according to the ISO/IEC 9796 series (option 1)

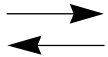
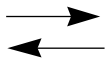
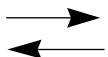
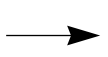
6.3.3 Building the DSI on ECDSA

No hash shall be internally computed by the ICC. The size of the DSI shall not be greater than the size of the order of the base point (this point is relevant in particular for elliptic curves whose prime length is not a multiple of eight bits – e.g. P-521).

6.4 Client/Server protocol

Table 2 shows the execution flow of the RSA client/server authentication. This specification covers only the internal authentication.

Table 2 — Client/Server authentication flow

Stage	Step	IFD	Trans- mission	ICC
1	—	... preceding steps(e.g. user verification) that are application specific for client/server authentication		application specific protocol flow
INTERNAL AUTHENTICATION — server (IFD) authenticates client (ICC)				
2	1	READ BINARY of certificate (typically X.509)		certificate
	2	MSE:SET PrK.CH.AUT		
	3	INTERNAL AUTHENTICATE or COMPUTE DIGITAL SIGNATURE		
Client (ICC) is authenticated to server (IFD)				
3	—	... subsequent steps that are application specific to present client/server authentication		not specified in this document

6.4.1 Step 1 — Read certificate

If the IFD does not already have the client's (ICC) authentication certificate, it may read it from a global data file EF.C.CH.AUT. The client's certificate C.CH.AUT may contain public algorithm quantities which depend on the client/server authentication algorithm.

Execution Flow

Table 3 — Read EF.C.CH.AUT — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'B0' READ BINARY
P1	'85' Example: SFI = 05
P2	'00' offset = 0
Le field	'00'

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.2.3.

The structure and coding of C.CA_{ICC}.CS_AUT may be as described in section 15.13 "C_CV.CA.CS-AUT (non self-descriptive)", but it could also be some other certificate format unknown to the E_{SIGN} application. The actual format is application specific and out of the scope of this standard.

Table 4 — Read EF.C.CH.AUT — response

Response Parameter	Meaning
Data field	EF.C.CH.AUT plain data as stored in EF
SW1-SW2	Refer to ISO/IEC 7816-4

Unless the READ BINARY selects the file using a Short File Identifier (SFI), an appropriate SELECT(EF) command shall be executed prior to reading the file.

6.4.2 Step 2 — Set signing key for client/server internal authentication

The MSE command sets the keyID of the ICC's client/server private authentication key to be used for the following computation of the digital signature.

Execution Flow

Table 5 — Select key reference for internal authentication - command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'41' SET for internal authentication
P2	'A4'
Lc field	'xx'
Data field	CRT data field with AlgID (PrK.CH.AUT) and KeyRef(PrK.CH.AUT)

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.5.11.

The structure and coding of the key reference data is specified in section 12.2.2 "CRT AT for selection of ICC's private client/server auth. key".

In order to avoid ambiguity the hashing function MGF1 shall be indicated in the AlgID.

The response data field is empty.

Table 6 — Select key reference for internal authentication — response

Response Parameter	Meaning
Data field	empty
SW1-SW2	Refer to ISO/IEC 7816-4

Alternatively the COMPUTE DIGITAL SIGNATURE command can be used.

Alternative Execution Flow

Table 7 — Select key reference for internal authentication — command APDU

Command Parameter	Meaning
CLA	according to ISO 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'41' SET for internal authentication
P2	'B6' digital signature DST
Lc field	'xx'
Data field	CRT data field with AlgID (PrK.CH.AUT) and KeyRef(PrK.CH.AUT)

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.5.11.

The structure and coding of the key reference data is specified in section 11.2.1 "CRT DST for selection of ICC's private client/server auth. key".

The response data field is empty.

Table 8 — Select key reference for internal authentication — response

Response Parameter	Meaning
Data field	empty
SW1-SW2	Refer to ISO/IEC 7816-4

6.4.3 Step 3 — Internal authentication

The IFD sends a challenge in the data field of the INTERNAL AUTHENTICATE command. The ICC computes a signature over the challenge by using its private key, that was selected in the preceding step.

Execution Flow

Table 9 — INTERNAL AUTHENTICATE — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'88' INTERNAL AUTHENTICATE
P1	'00' no algorithm information
P2	'00' key reference — none
Lc field	'xx'
Data field	data authentication input see below (Table 13 to Table 17)
Le field	'00'

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.5.2.

The digital signature input format is described in 6.4.5 "Command data field for the client server authentication".

The algorithms usable for client server authentication are described in 12.1 "Algorithm Identifiers".

The response data contains the signature.

Table 10 — INTERNAL AUTHENTICATE — response

Response Parameter	Meaning
Data field	RSA: $SIG = DS^a[PrK.CH.AUT](DSI)$ where DSI is formed according 6.3 "Padding format". ECDSA: For the ECDSA algorithm the signature format is described in EN 419212-1:2014, 14.3.6.
SW1-SW2	Refer to ISO/IEC 7816-4
^a DS is to be understood as plain RSA exponentiation. It does not imply a layer of input formatting other than already specified here.	

Alternatively, the COMPUTE DIGITAL SIGNATURE command can be used.

Alternative Execution Flow

Table 11 — COMPUTE DIGITAL SIGNATURE- command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'2A' PERFORM SECURITY OPERATION
P1	'9E' expected response DO
P2	'9A' data to be signed
Lc field	'xx'
Data field	data authentication input T see below
Le field	'00'

For the structure and content of the APDU refer to ISO/IEC 7816-8:2004, 5.2.

The digital signature input format is described in 6.4.5 "Command data field for the client server authentication".

The algorithms usable for client server authentication are described in 12.1 "Algorithm Identifiers".

The response data contains the signature.

Table 12 — COMPUTE DIGITAL SIGNATURE — response

Response Parameter	Meaning
Data field	RSA: $SIG = DS^a[PrK.CH.AUT](DSI)$ where DSI is formed according to 6.3 "Padding format". ECDSA: For the ECDSA algorithm the signature format is described in EN 419212-1:2014, 14.3.6.
SW1-SW2	Refer to ISO/IEC 7816-4
^a DS is to be understood as plain RSA exponentiation. It does not imply a layer of input formatting other than already specified here.	

In case of RSA, while the IFD sends only the authentication input T, the ICC performs the padding as described in Figure 4. In the following text it is not necessary to distinguish between these alternatives.

6.4.4 Client/Server authentication execution flow

The ICC generates a signature over the input data, which shall be presented in the response data. For RSA the plain signature to be presented is the exponentiation result in the same size as the modulus.

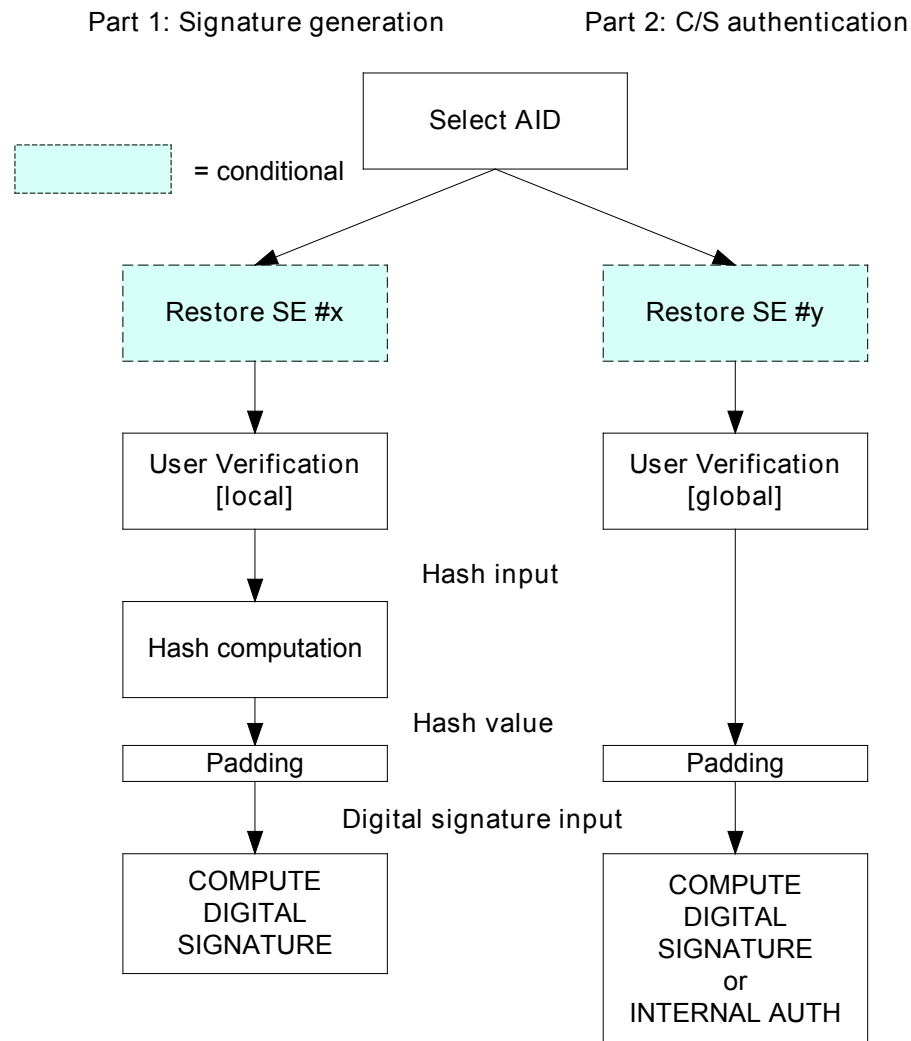


Figure 7 — Comparison of operations in the ICC for signature creation versus C/S auth. (informative)

Figure 7 compares the signature generation process with a Client/Server (C/S) authentication. On Client/Server authentication the following rules shall apply:

- The selection of a security environment is mandatory if the COMPUTE DIGITAL SIGNATURE command is used for signature generation and client/server authentication. Using different commands avoids ambiguity when using COMPUTE DIGITAL SIGNATURE, hence a selection of a security environment is not necessary.
- For C/S authentication the authentication input is generated by the PC and sent to the ICC. The ICC computes the digital signature. If padding is involved at least part of the padding needs to be done in the ICC in order to avoid existing attacks on the keys.
- The digest info is included in the data field of the COMPUTE DIGITAL SIGNATURE command. This mechanism is different from the computation of a digital signature as described in EN 419212-1:2014, 7.4.
- If user verification is required, it shall be performed with password references different for the signature generation [local password] and for C/S authentication [global password].

6.4.5 Command data field for the client server authentication

6.4.5.1 RSA

Table 13 — Command data field for KERBEROS protocol

T	L	V	Bytes
—	—	DigestInfo acc. to [3]	

Table 14 — Command data field for SSL3.0 [20], TLS1.0 [4] and TLS 1.1 [5]

T	L	V	Bytes
—	—	H_MD5 H_SHA1 refer to [7] and [8].	

Table 15 — Command data field for SSH

T	L	V	Bytes
—	—	T = H_SHA1 refer to [19]	

Table 16 — Command data field for TLS 1.2

T	L	V	Bytes
—	—	DigestInfo acc. to [6]	

The ICC shall perform the padding according to 6.3 "Padding format". The authentication input is taken as described by the three tables above.

6.4.5.2 ECDSA

For ECDSA the authentication input is padded with leading zero bits. The build scheme for the signature is described in 6.3.3 "Building the DSI on ECDSA".

6.4.5.3 Other algorithms

All other algorithms use the padded (PKCS #1) authentication input in the data field.

Table 17 — Input to other algorithms

T	L	V	Bytes
—	—	authentication input	

7 Role Authentication

7.1 Role Authentication of the card

This feature is optional.

The C/S authentication is based on an Internal Authentication protocol whereby the server (IFD) authenticates the card (ICC). For this purpose, the IFD reads the card's certificate (usually a X.509 certificate) that may provide extension (acc. X.509) denoting the role with which the cardholder is endowed to access some reserved data hosted on the server. The differential access to these data is controlled by the server according to the rules defined by the certificate extensions. This role authentication of the card is application-specific and not described further in this document.

7.2 Role Authentication of the server

Asymmetric role authentication applying Elliptic Curve Cryptography and Diffie Hellman key exchange are already specified in EN 419212-1, i.e. device authentication with privacy (8.7 in EN 419212-1) and mEAC protocol (8.8 in EN 419212-1:2014).

With the successful execution and establishment of a secure messaging channel the protocols provide role authentication as an integral part of the import of CV certificates. Role identifier values and access conditions are coded in the CHA (15.7.4 in EN 419212-1:2014) or CHAT object (15.7.6 in EN 419212-1:2014). As a consequence role authentication is an optional feature in these cases.

This clause describes the procedure to authenticate an external entity to the card. The authentication associates a specific role (e.g. access rights) to the entity. This clause supports these two mechanisms:

- A symmetric role authentication (external authentication).
- An asymmetric role authentication.

The following procedure describes:

- The cryptographic operation that supports the authentication.
- The specification of the associated role in the card.

7.3 Symmetrical external authentication

7.3.1 Protocol

The authentication procedure is based on a symmetric key set using TDES or AES.

7.3.1.1 Keys definition

The authentication procedure involves a symmetric key set.

In case the TDES scheme is used, the key set is made of two TDES EDE2 keys:

KS.KENC: for confidentiality — 16 bytes long.

KS.KMAC: for integrity (MAC) — 16 bytes long.

In case AES scheme with CMAC is used, the key set is made of two AES keys:

KS.KENC: for confidentiality — 16, 24 or 32 bytes long.

KS.KMAC: for integrity (MAC) — 16, 24 or 32 bytes long.

In case AES scheme with EMAC is used, the key set is made of three AES keys:

KS.KENC: for confidentiality — 16, 24 or 32 bytes long.

KS.KMAC1: for integrity (MAC) — 16, 24 or 32 bytes long.

KS.KMAC2: for integrity (MAC) — 16, 24 or 32 bytes long.

For the AES key sets, all keys in a set shall be of same length (16, 24 or 32 bytes).

7.3.1.2 Naming rules

ENC Encryption with TDES or AES in CBC mode and initial vector IV = '00 ... 00' using KS.KENC.

AlgoMAC Integrity Cryptogram (MAC) with KS.KMAC.

SN.ICC:8 ICC serial number, 8 least significant bytes

RND.ICC Random number (of length equal to the block size) generated by the ICC and used to authenticate the external entity.

At the end of this sequence of commands the IFD is authenticated.

Table 18 — Symmetric external authentication

Step	IFD	Trans- mission	ICC
1	READ BINARY (EF.SN)	→ ←	Return SN.ICC Card Serial Number in the context of the current application
2	MSE: Set AT	→ ←	Key & algorithm selection (unless implicitly known or specified in step 4)
3	GET CHALLENGE	→ ←	Return random number RND.ICC
4	EXTERNAL AUTHENTICATE	→ ←	MAC checking Decipher Verify RND.ICC (IFD is authenticated)

7.3.1.3 Step 1 — Read key exchange parameters

The IFD requires the ICC's serial number in order to derive the appropriate keys.

Execution Flow

Table 19 — Read EF.SN.ICC parameters — command APDU

Command Parameter	Meaning
CLA	according to ISO 7816-4
INS	'B0' READ BINARY
P1	'85' Example: SFI = 05
P2	'00' offset = 0
Le field	'00'

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.2.3.

The contents of EF.SN are specified in 16.4 "EF.SN.ICC" of EN 419212-1:2014 .

Table 20 — Read EF.SN.ICC — response

Response Parameter	Meaning
Data field	content of EF.SN.ICC
SW1-SW2	refer to ISO/IEC 7816-4

Unless the READ BINARY selects the file using a Short File Identifier (SFI), an appropriate SELECT(EF) command shall be executed prior to reading the file. This also applies to the following steps, where appropriate.

7.3.1.4 Step 2 — Select Key for symmetrical external authentication

The IFD selects the keys and the algorithms to be used for the external authentication. If the IFD can be sure, that the ICC knows the keys/algorithms implicitly by the nature of the application, step 2 may be skipped.

Table 21 — Select Key for symmetrical external authentication — command APDU

Command Parameter	Meaning		
CLA	according to ISO/IEC 7816-4		
INS	‘22’	MANAGE SECURITY ENVIRONMENT	
P1	‘81’	MSE SET	
P2	‘A4’	AT (authentication template)	
Lc field	‘xx’		
data	CRT data field as follows		
	‘80’	‘xx’	algorithm identifier acc. to EN 419212-1:2014, Annex A
	‘83’ (CONDITIONAL) ¹	‘xx’	Key identifier

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.5.11. The structure and coding of the key reference data is specified in EN 419212-1:2014, 12.2.2. The response data field is empty.

¹ If the keyID of the keys KS.KENC/KS.KMAC will be set by the EXTERNAL AUTHENTICATE (Table 25) then the key identifier DO '83' is not present.

Table 22 — Select Key for symmetrical external authentication — response APDU

Response Parameter	Meaning
Data field	Empty
SW1-SW2	refer to ISO/IEC 7816-4

This step is used to load the context to use for the authentication procedure. Alternatively this context may be loaded from a SE stored in the ICC with the command MSE RESTORE.

7.3.1.5 Step 3 — Challenge generation

The IFD requests a challenge from the IFD. Depending on the type of cryptography considered, the length of challenge required by the IFD is either 8 bytes long (authentication based on TDES) or 16 bytes long (authentication based on AES).

Table 23 — GET CHALLENGE — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'84' GET CHALLENGE
P1	'00'
P2	'00'
Le field	< cipher block size >

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.5.3.

The response data contains the challenge.

Table 24 — GET CHALLENGE — response

Response Parameter	Meaning
Data field	S = RND.ICC < cipher block size >
SW1-SW2	Refer to ISO/IEC 7816-4

7.3.1.6 Step 4 — External authentication

The IFD computes a signature on the challenge using the ICC's (derived) secret key KS.KENC. After the ICC verifies the signature with the same key, it responds to confirm whether the authentication was successful or not.

Execution Flow**Table 25 — EXTERNAL AUTHENTICATE — command APDU**

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'82' EXTERNAL AUTHENTICATE
P1	'00' no specific information
P2	'xx' key reference denoting KS.KENC and KS.KMAC (CONDITIONAL) ¹
Lc field	'xx'
data	Authentication token μ as described in 7.3.2.

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.5.4.

¹ If the keyID of the keys KS.KENC/KS.KMAC (KS.KMAC1 / KS.KMAC2) was set by using an MSE command prior to EXTERNAL AUTHENTICATE then P2 = '00'.

Table 26 — EXTERNAL AUTHENTICATE — response

Response Parameter	Meaning
Data field	empty
SW1-SW2	Refer to ISO/IEC 7816-4

7.3.2 Description of the cryptographic mechanisms

The authentication relies on the computation of an authentication token computed from

$$\mathbf{S} = \text{RND.ICC} \parallel \text{SN.ICC:8 (8 LSB)}$$

The authentication token μ is computed from $\mathbf{S}$ as follows

If TDES scheme is used ·

- $\mathbf{S}$ is encrypted to get $\text{ENC}(\mathbf{S})$ using 3DES in CBC mode with $\text{IV} = "00 \dots 00"$ (8 Bytes) under the Key KS.KENC without any padding, as described in 10.9.2 of EN 419212-1:2014.
- A Checksum is computed over $\text{ENC}(\mathbf{S})$ to get $\text{MAC}(\mathbf{S})$ under the Key KS.KMAC as described in 10.9.2 of EN 419212-1:2014. The IV used during computation is $\text{IV} = "00 \dots 00"$ (8 Bytes).

If AES scheme is used (case with CMAC)

- $\mathbf{S}$ is encrypted to get $\text{ENC}(\mathbf{S})$ using AES in CBC mode with $\text{IV} = "00 \dots 00"$ (16 Bytes) under the key KS.KENC as described in 10.9.2 of EN 419212-1:2014. Padding method 2 according to ISO/IEC 9797-1 is applied.
- A Checksum is computed over $\text{ENC}(\mathbf{S})$ to get $\text{MAC}(\mathbf{S})$ under the Key KS.KMAC as described in 10.9.3 of EN 419212-1:2014. The IV used during computation is $\text{IV} = "00 \dots 00"$ (16 Bytes).

If AES scheme is used (case with EMAC)

- $\mathbf{S}$ is encrypted to get $\text{ENC}(\mathbf{S})$ using AES in CBC mode with $\text{IV} = "00 \dots 00"$ (16 Bytes) under the key KS.KENC as described in 10.9.2 of EN 419212-1:2014. Padding method 2 according to ISO/IEC 9797-1 is applied.
- A checksum is computed over $\text{ENC}(\mathbf{S})$ to get $\text{MAC}(\mathbf{S})$ under the Keys KS.KMAC1 and KS.KMAC2 as described in 10.9.4 of EN 419212-1:2014. The IV used during computation is $\text{IV} = "00 \dots 00"$ (16 Bytes).

The authentication token μ is made up with $\text{ENC}(\mathbf{S}) \parallel \text{MAC}(\mathbf{S})$

$$\mu = \text{ENC}(\mathbf{S}) \parallel \text{MAC}(\mathbf{S})$$

7.3.3 Role description

The card implicitly knows the role associated with the *KeySet*. Prior to granting privileges associated with the role, the card validates the cryptogram delivered by the IFD as described above.

7.4 Asymmetric external authentication

Public keys are transported through Card Verifiable Certificates (CVC).

7.4.1 Protocol based on RSA

Notation:

CERT.PATH	Certificate chain constructed by the entity.
PrK.IFD.RA	IFD's role authentication private key
PuK.IFD.RA	IFD's role authentication public key
RND. ICC	Card random number — 8 bytes.
SN.ICC:8	Card serial number — 8 least significant bytes.
DS	Digital Signature algorithm as specified by ISO/IEC 9796-2 scheme 1.
DS⁻¹	Digital Signature Verification algorithm as specified by ISO/IEC 9796-2 scheme 1.

Table 27 — External authentication based on RSA

Step	IFD	Trans- mission	ICC
Prepare the CERT.PATH			
1	Successive certificate verification until the final CVC is available	→ ←	Step by step, chain verification. Finally the PuK.IFD.RA and other associated parameters are known by the ICC.
2	MSE (KeyId, AlgId) Select PuK.IFD.RA for external authentication	→ ←	Key Selection
3	GET CHALLENGE	→ ←	Generate Random number RND.ICC
4	External entity authentication EXTERNAL AUTHENTICATE	→ ←	DS ⁻¹ (SIG): Signature verification with PuK.IFD.RA

7.4.1.1 Step 1 — Success certificate verification

The verification of certificates along a certificate chain is similar to those described in the device authentication in EN 419212-1.

7.4.1.2 Step 2 — Selection of verification key PuK.IFD.RA

The MSE command sets the keyID of the CA's public key for role authentication.

Execution Flow

Table 28 — Select key reference for role authentication — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'81' SET for verification
P2	'A4' AUT
Lc field	'xx'
Data field	'83' L _{CHR} PuK.IFD.RA Key reference for role authentication verification '80 01' <algorithm identifier for asymmetric external authentication> (refer to Annex C)

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.5.11.

The response data field is empty.

Table 29 — Select key reference for certificate verification — response

Response Parameter	Meaning
Data field	empty
SW1-SW2	Refer to ISO/IEC 7816-4

7.4.1.3 Step 3 — Get Challenge

In order to prove its authenticity dynamically the IFD shall request a challenge from the ICC. The challenge may consist of a random number.

Execution Flow

Table 30 — GET CHALLENGE — command APDU

Command Parameter	Meaning
CLA	according to ISO 7816-4
INS	'84' GET CHALLENGE
P1	'00'
P2	'00'
Le field	'08'

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.5.3.

The response data contains the 8 byte challenge.

Table 31 — GET CHALLENGE — response

Response Parameter	Meaning
Data field	RND.ICC 8 byte random number
SW1-SW2	Refer to ISO/IEC 7816-4

7.4.1.4 Step 4 — External authentication

The IFD computes a signature on the concatenation of the challenge using PrK.IFD.RA.

After the ICC verifies the signature with the trusted public key PuK.IFD.RA, it responds to confirm whether the authentication was successful or not.

Execution Flow

Table 32 — EXTERNAL AUTHENTICATE — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'82' EXTERNAL AUTHENTICATE
P1	'00'
P2	'00'
Lc field	'xx'
data	SIG = DS ^a [PrK.IFD.RA]('6A' PRND2 h(PRND2 RND.ICC SN.ICC 'BC') where the hash function h() is SHA-1 PRND2 = 'xx..xx' random padding bytes generated by IFD. PRND2 shall be such a length that the term '6A' – 'BC' has the same length as the modulus.
^a DS is to be understood as plain RSA exponentiation. It does not imply a layer of input formatting other than already specified here.	

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.5.4.

Table 33 — EXTERNAL AUTHENTICATE — response

Response Parameter	Meaning
Data field	empty
SW1-SW2	Refer to ISO/IEC 7816-4

7.4.1.5 Role description

The role is described in the CHA field of the final CVC. After a successful authentication, the ICC authorizes access to any data protected by the role retrieved from the certificate.

7.4.2 Protocol based on modular Enhanced Role Authentication (mERA)

mERA is a modular role authentication protocol. It can be used for mERA-based eServices with a trusted third party protocol, to enable the IFD or the service provider (IFD/SP) to authenticate to the ICC:

- once, before the *IFD/SP* is authorized to write the set of criteria onto the ICC (mERA stages 1-3),
- once, before the *IFD/SP* is authorized to read the privacy-preserving credentials from the ICC (mERA stages 1-7)

A security proof is given in [16] and [17]. The following notation is used in particular with the mERA protocol

Application-specific payload

Data specific to the application (e.g. public key, criteria list or Auxiliary data...), may additionally comprise a descriptor. This data takes two different values in the following protocol depending on the processed steps:

- in case of mERA1-3, the application-specific payload corresponds to the list of criteria asked by the SP (also called "Profile" in 11.4.1 "mERA-based eServices with trusted third party protocol").
- in case of mERA1-7, the application-specific payload corresponds to the public part of the ephemeral key generated by ICC (PuK.ICC.KA)

MSK.ICC	Master secret key for authentication owned by the ICC;
EPK.IFD	Intermediate result of an encryption as described in the associated section;
SK.IFD	Authentication secret key of the IFD;
SN.IFD	Serial number of the IFD – it is used to build SK.IFD from MSK.ICC;
RND1.IFD	Random number generated by the IFD – it is used to create a session key from SK.IFD
RND.ICC	Random number generated by the ICC – it is used to create a session key from SK.IFD
$K_{ENC}^a, K_{ENC}^b, K_{MAC}$	Session keys for authentication – they are derived from SK.IFD, RND1.IFD and RND.ICC;
AlgoID	Algorithm identifier indicating the flavour of mERA to use;
ENC	Encryption algorithm associated to the flavour of mERA;
MAC	Checksum computation algorithm associated to the flavour of mERA;
MPK	Intermediate result of a cryptographic checksum over EPK;
KDF	Key Derivation function associated to the flavour of mERA;
KA	Key Agreement (may be DH or ECDH);
CRT	Control Reference Template
h	Hashing function associated to the flavour of mERA;
RND2.IFD	Random number generated by the IFD – It is used to authenticate the ICC;
PuK.IFD.KA, PrK.IFD.KA	ephemeral key pair for key agreement of the IFD;
PuK.ICC.KA, PrK.ICC.KA	ephemeral key pair for key agreement of the ICC;
PuK.IFD.GENUINE, PrK.IFD.GENUINE	static asymmetric key pair for checking that the IFD is genuine;
D	Domain parameters used by the ICC. They match the asymmetric key pairs (PuK.IFD.KA, PrK.IFD.KA) & (PuK.ICC.KA, PrK.ICC.KA). DH or ECDH may be used;