



BSI Standards Publication

Application Interface for Secure Elements for Electronic Identification, Authentication and Trusted Services

Part 3: Device authentication protocols

National foreword

This British Standard is the UK implementation of EN 419212-3:2017. Together with BS EN 419212-1:2017, BS EN 419212-2, BS EN 419212-4, and BS EN 419212-5, it supersedes BS EN 419212-1:2014 and BS EN 419212-2:2014, which will be withdrawn upon publication of the rest of the series.

The UK participation in its preparation was entrusted to Technical Committee IST/17, Cards and personal identification.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2017
Published by BSI Standards Limited 2017

ISBN 978 0 580 95789 5

ICS 35.240.15

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 September 2017.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

English Version

Application Interface for Secure Elements for Electronic Identification, Authentication and Trusted Services - Part 3: Device authentication protocols

Interface applicative des éléments sécurisés utilisés
comme dispositifs de création de signature
électronique qualifiée (cachet) Partie 3: Protocoles
d'authentification des dispositifs

Anwendungsschnittstelle für Smartcards als sichere
Signaturerstellungseinheiten - Teil 3:
Geräteauthentisierungsprotokolle

This European Standard was approved by CEN on 17 March 2017.

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, Serbia, 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

European foreword.....	5
Introduction	6
1 Scope	7
2 Normative references.....	7
3 Device authentication.....	7
3.1 General.....	7
3.2 Asymmetric Authentication introduction.....	9
3.3 Certification authorities and certificates	9
3.3.1 Certificate chains.....	9
3.3.2 Usage of link certificates.....	10
3.4 Authentication environments	10
3.4.1 SCA in trusted environment	11
3.4.2 SCA in untrusted environment.....	11
3.4.3 Specification of the environment	11
3.4.4 Display message mechanism	11
3.4.5 Additional authentication environments.....	12
3.5 Key transport and key agreement mechanisms	12
3.6 Device authentication with privacy protection.....	12
3.6.1 General.....	12
3.6.2 Authentication steps	13
3.7 Privacy constrained Modular EAC (mEAC) protocol with non-traceability feature	31
3.7.1 General.....	31
3.7.2 Example for traceability case.....	31
3.7.3 Notation	32
3.7.4 Authentication steps	32
3.7.5 Unlinkability Mechanism with individual private keys	45
3.8 Symmetric authentication scheme	54
3.8.1 General.....	54
3.8.2 Authentication steps	54
3.8.3 Session Key creation	58
3.9 Key transport protocol based on RSA.....	58
3.9.1 General.....	58
3.9.2 Authentication Steps.....	60
3.9.3 Session Key creation	68
3.10 Compute Session keys from key seed $K_{\text{IFD/ICC}}$	68
3.10.1 General.....	68
3.10.2 Generation of key data	69
3.10.3 Partitioning of the key data.....	69
3.10.4 Algorithm and method specific definition for key derivation	69
3.10.5 Key derivation from passwords.....	72
3.11 Compute send sequence counter SSC.....	73
3.12 Post-authentication phase	73
3.13 Ending the secure session	74
3.13.1 General.....	74
3.13.2 Example for ending a secure session	74
3.13.3 Rules for ending a secure session	74

3.14	Reading the Display Message	75
3.15	Updating the Display Message	77
4	Data structures	78
4.1	General	78
4.2	CRTs.....	78
4.2.1	General	78
4.2.2	CRT AT for the selection of internal private authentication keys.....	78
4.2.3	CRT AT for selection of internal authentication keys.....	78
4.2.4	CRT for selection of IFD's PuK.CA _{IFD} .CS_AUT	79
4.2.5	CRT for selection of IFD's PuK.IFD.AUT	79
4.2.6	CRT AT for selection of the public DH / ECDH key parameters	80
4.2.7	GENERAL AUTHENTICATE DH key parameters used by the Privacy Protocol	80
4.2.8	CRT AT for selection of ICC's private authentication key.....	80
4.2.9	CRT for selection of IFD's PuK.IFD.AUT	81
4.2.10	CRT for selection of PrK.ICC.KA	81
4.3	Key transport device authentication protocol.....	82
4.3.1	EXTERNAL AUTHENTICATE	82
4.3.2	INTERNAL AUTHENTICATE	82
4.4	Privacy device authentication protocol.....	83
4.4.1	EXTERNAL AUTHENTICATE (DH case)	83
4.4.2	EXTERNAL AUTHENTICATE (ECDH case)	84
4.4.3	INTERNAL AUTHENTICATE (DH case).....	85
4.4.4	INTERNAL AUTHENTICATE (ECDH case).....	85
5	CV_Certificates and Key Management	86
5.1	General	86
5.2	Level of trust in a certificate	86
5.3	Key Management.....	86
5.4	Certificate types	87
5.4.1	Card Verifiable Certificates	87
5.4.2	Signature-Certificates	88
5.4.3	Authentication Certificates	88
5.5	Use of the public key extracted from a CV-certificate.....	88
5.6	Validity of the key extracted from a CV-certificate.....	88
5.7	Structure of CVC.....	89
5.7.1	General	89
5.7.2	Non-self-descriptive certificates	89
5.7.3	Self-descriptive certificates	90
5.8	Certificate Content.....	90
5.8.1	General	90
	CPI-Certificate Profile Identifier.....	91
5.8.2	CAR-Certification Authority Reference DO.....	92
5.8.3	CHR-Certificate Holder Reference DO	93
5.8.4	CHA-Certificate Holder Authorization Data Object (CHA-DO).....	94
5.8.5	Role identifier specifications.....	95
5.8.6	User and service provider authentication.....	97
5.8.7	CHAT-Certificate Holder Authorization Template (CHAT).....	98
5.8.8	OID — Object identifier	98
5.8.9	CEDT — Certificate Effective Date Template	98
5.8.10	CXDT — Certificate Expiration date Template.....	98
5.9	Certificate signature	99
5.9.1	General	99
5.9.2	Non self-descriptive certificates.....	99
5.9.3	Self-descriptive certificates	100

5.10	Coding of the certificate content.....	101
5.10.1	Non self-descriptive certificates	101
5.10.2	Self-descriptive certificates.....	101
5.10.3	Self-descriptive certificates for elliptic curve cryptography.....	102
5.11	Steps of CVC verification	105
5.11.1	General.....	105
5.11.2	First round: CVC verification from a Root PuK.....	106
5.11.3	Subsequent round(s)	107
5.12	Commands to handle the CVC	107
5.13	C_CV.IFD.AUT (non self-descriptive)	107
5.14	C_CV.CA.CS-AUT (non self-descriptive).....	108
5.15	C.ICC.AUT.....	109
5.16	Self-descriptive CV Certificate (Example)	110
5.16.1	General.....	110
5.16.2	Public Key	110
5.16.3	Certificate Holder Authorization Template.....	111
5.16.4	Certificate Extension	111
5.16.5	ECDSA Signature	112
	Annex A (informative) Device authentication Protocol Properties.....	113
	Bibliography.....	115

European foreword

This document (EN 419212-3:2017) has been prepared by 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 March 2018, and conflicting national standards shall be withdrawn at the latest by March 2018.

This document supersedes EN 419212-1:2014 and EN 419212-2:2014.

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

This standard supports services in the context of electronic IDentification, Authentication and Trust Services (eIDAS) including signatures.

In EN 419212 Part 2, the standard allows support of implementations of the European legal framework for electronic signatures, defining the functional and security features for a Secure Elements (SE) (e.g. smart cards) intended to be used as a Qualified electronic Signature Creation Device (QSCD) according to the Terms of the “European Regulation on Electronic Identification and Trust Services for electronic transactions in the internal market” [1].

A Secure Element (SE) compliant to the standard will be able to produce a “qualified electronic signature” that fulfils the requirements of Article of the Electronic Signature Regulation [1] and therefore can be considered equivalent to a hand-written signature.

This standard consists of five parts:

Part 1: “Introduction and common definitions” describes the history, application context, market perspective and a tutorial about the basic understanding of electronic signatures. It also provides common terms and references valid for the entire 419212 series. [24]

Part 2: “Signature and Seal Services” describes the specifications for signature generation according to the eIDAS regulation. [25]

Part 3: “Device Authentication” describes the device authentication protocols and the related key management services to establish a secure channel. [26]

Part 4: “Privacy specific Protocols” describes functions and services to provide privacy to identification services. [27]

Part 5: “Trusted eServices” describes services that may be used in conjunction with signature services described in Part 2. [28]

According to the CEN-CENELEC Internal Regulations, the national standards organisations 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, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

Introduction

The European Committee for Standardization (CEN) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent concerning the mapping function given in [25] 8.2.5 “Step 4.2 - Map nonce and compute generator point for integrated mapping”.

The patent relates to “Sagem, MorphoMapping Patents FR09-54043 and FR09-54053, 2009”.

CEN takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has ensured CEN that he/she is willing to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with CEN. Information may be obtained from:

Morpho

11, boulevard Galliéni

92445 Issy-les-Moulineaux Cedex

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. CEN shall not be held responsible for identifying any or all such patent rights.

1 Scope

This part specifies device authentication to be used for QSCDs in various contexts including:

- Device authentication protocols;
- Establishment of a secure channel;
- Data structures;
- CV-certificates;
- Key management.

The device authentication protocols should apply to sole-control signature mandated by the EU-regulation eIDAS [1].

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.

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

ISO/IEC 7816-6, *Identification cards — Integrated circuit cards — Part 6: Interindustry data elements for interchange*

ISO/IEC 7816-8:2004, *Identification cards — Integrated circuit cards — Part 8: Commands for security operations*

ISO/IEC 9796-2:2010, *Information technology — Security techniques — Digital signature schemes giving message recovery — Part 2: Integer factorization based mechanisms*

ISO/IEC 14888-3:2016, *Information technology — Security techniques — Digital signatures with appendix — Part 3: Discrete logarithm based mechanisms*

3 Device authentication

3.1 General

This clause assumes that device authentication has to be performed as required in 3.3.

Device authentication requires mandatory steps in order to provide a secure authentication. A device authentication is mutual and combines two mechanisms:

- an ICC verifies the external world (TDA) and itself verified by the external world (CDA);
- the two devices negotiate or exchange information to establish common symmetric session keys for subsequent operations.

After negotiation of the symmetric keys, a secure session is established. A secure session is a cryptographic protection of the messages from both sides. The cryptographic protection can be:

- a cryptographic checksum on a plain text for integrity protection and/or
- an encrypted message text for confidentiality protection (mandates cryptographic checksum).

Refer to “9 Secure Messaging” [25].

For performance reasons, the secure messaging keys are symmetric keys. Therefore this document describes the establishment of symmetric session keys only, and does not consider an option for asymmetric session keys.

Once the session keys are established, a trusted channel is available to protect or conceal the information transmitted over the interface. The application of Secure Messaging (SM) is **mandatory** for **every subsequent operation** to ensure the provision of an entirely trusted channel. For exceptions refer to “5.3 Selection of ESIGN application” [25]. The conditions to end a secure messaging session are given in 3.12.

This chapter describes the following device authentication mechanisms

- asymmetric session key transport mechanism based on RSA;
- asymmetric session key agreement mechanism with privacy protection;
- asymmetric session key agreement mechanism with non-traceability and privacy protection;
- symmetric authentication mechanism

in order to perform a mutual authentication protocol between IFD and ICC. The presentation of the asymmetric schemes is sequentially staged in order to trade off the security features with the required complexity of the authentication protocols.

NOTE Authentication in general requires a defined order of steps to be processed. Violation of this order may result in the ICC aborting the process. The mechanism to control the proper order of execution steps are out of the scope of this standard.

In order to distinguish which device authentication is the most appropriate for a given situation refer to Annex A.

For the use of certificates the following table indicates the correct certificate type for each device authentication protocol.

Table 1 — Certificate type for use with device authentication protocols

Certificate Device Auth.	CV Certificate self-descriptive	CV Certificate non descriptive	self- Attribute certificates self-descriptive
3.8 Key transport protocol	-	x	-
3.5 Privacy protocol RSA	-	x	-
3.5 Privacy protocol ELC	x	-	x
3.6 mEAC	x	-	x

3.2 Asymmetric Authentication introduction

The above steps result in a high level of security for device authentication. They contain some mandatory aspects of the device authentication protocols considered to comply with the claims of the ESIGN-G1 [23] document.

- The public key of the IFD shall not be used until successfully verified by a certificate.
- The device to be authenticated shall submit its identity relevant token to the authenticator in a secure way, typically this can be achieved implicitly if the identity is part of the certificate.
- For both sides of the authentication it is mandatory to have evidence (after authentication), that the authentication was made with the same entity that shares the session keys for the secure messaging.
- After device authentication a trusted channel has been established. Relevant subsequent commands shall use the secure messaging session.
- A display message may be used to indicate successful device authentication in order to inform the card holder of the presence of a secure channel. The application of the display message is described in 3.13.

3.3 Certification authorities and certificates

3.3.1 Certificate chains

Asymmetric authentication requires the presentation of public keys in certificates. A chain of certificates might have to be presented. To establish the 'starting point of trust' the first certificate sent to the ICC shall always be verifiable with a public key already existing in the ICC. The most general of such keys available in the ICC can be a Root CA public key. Each certificate introduces another public key, while the presented certificate is always signed with a key, trusted by the ICC.

As the key pair of the Root CA changes over time, link certificates provide a link between expired and valid key pairs (refer to 3.2.2).

The Root CA provides

- certificates for the subordinate CAs
- link certificates in order to convey the public key(s).

In order to use an ESIGN application a mutual authentication between ICC and customer terminal is required by the security policy. As PK algorithms are to be employed ICC and IFD authentication certificates are required.

The ICC is the carrier of the ICC authentication certificate; carrier of the IFD authentication certificate could be a plug-in ICC as an example for the realization of an authentication module inside the Signature Creation Application ([SCA](#)).

The handling of the DOs 5.7.9 and 5.7.10 are out of the scope of this standard.

3.3.2 Usage of link certificates

A typical use of link certificates is to provide a link between expired and valid Root-CA key pairs.

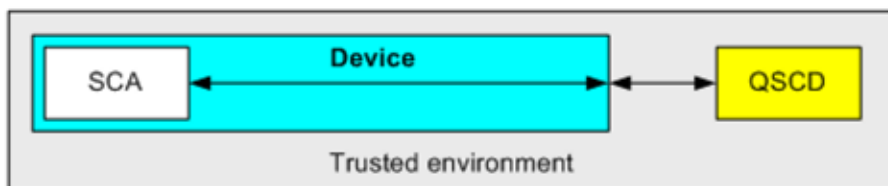


Figure 1 — Usage of link certificates for different key versions

The example above shows PuK.CA.AUT(2002) available at the ICC. This key was brought to the ICC in 2002, however in 2003 the root key was changed and another key chain

C.RCA.AUT(2003) → C.CA.AUT(2003) → C.IFD.AUT(2003)

was used when generating the IFD's certificate C.IFD.AUT(2003).

In order to allow authentication of the IFD, in the above example the ICC might have to receive a link certificate LC.RCA.AUT(2003) first. This certificate is signed with the PrK.RCA.AUT(2002) and it transports the PuK.RCA.AUT(2003).

After having verified the link certificate LC.RCA.AUT(2003) the ICC may now receive the certificates of the 2003-chain up to the C.IFD.AUT which carries the final public key required for authentication.

If it can verify the incoming certificate with an existing public key, the ICC accepts the transported public key and may use it for the next instance of the chain. The 'start of trust' is always established with the first certificate, being verified with a public key that is available in the ICC.

The example in Figure 1 considers a static Root CA key stored in the ICC. Hence, an IFD shall present a proper link certificate in every device authentication session.

Alternatively, an ICC may provide the possibility to permanently store the new Root CA key, which would require the import of a link certificate only once. If the trust anchor has been permanently updated within the ICC, the 'start of trust' for subsequent device authentications is the new Root CA key. The certificate chain to be presented by the IFD is shortened.

When verifying a certificate, the verifier shall verify the correct role of the signer. Only those roles being allowed to sign certificates may be accepted for the presentation of a certificate chain. The role of a certificate holder is specified in 5.7.4.

3.4 Authentication environments

NOTE In [22] two environments are distinguished with respect to signature creation applications.

3.4.1 SCA in trusted environment

The environment is considered to be trusted by the user. Device authentication is not required as the card holder knows the trusted environment that s(he) will apply the card to.

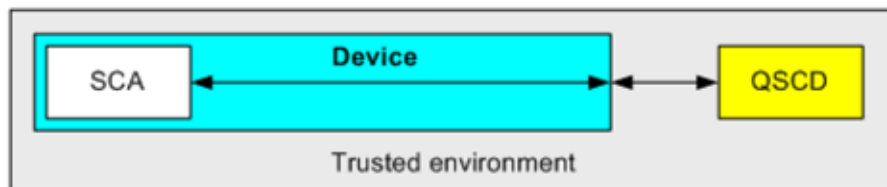


Figure 2 — Trust of the environment

3.4.2 SCA in untrusted environment

A device authentication shall be used if the operating environment of the ICC cannot be entirely trusted. This can be the case in public signature terminals or other devices, that cannot provide an a-priori secure channel.



Figure 3 — Communication in untrusted environment

Device authentication is mutual. The ICC shall authenticate the IFD and vice versa. The order of authentication, however, may differ, depending on the implemented scheme.

After successful device authentication, session keys are available on both sides to be used in subsequent transmissions. The appropriate secure messaging is in compliance with ISO/IEC 7816-4 and described in “9 Secure Messaging” [25].

Examples for an untrusted environment are:

- SCA and QSCD are not at the same location, i.e. the card is remote;
- usage of biometrics if the sensor is off-card;
- usage of contactless cards.

3.4.3 Specification of the environment

In general the IFD cannot evaluate a priori whether an environment is trustable or not. Therefore it shall be left to the card holder to decide, whether he may trust the application environment or not. The initiation of a device authentication in the application environment is out of the scope of this specification.

3.4.4 Display message mechanism

Given, the card holder has initiated a device authentication this specification proposes a 'display message' mechanism in order to verify that the device authentication has been performed successfully. If used, after successful device authentication, a display message shall indicate the successful device authentication in order to inform the user about the existence of a trusted channel. The application of the display message is described in 3.13.

3.4.5 Additional authentication environments

Device authentication may also be applied or necessary for multiple entity personalization (not in the scope of this standard).

3.5 Key transport and key agreement mechanisms

Key transport is the process of transferring a secret key, chosen by one entity (or a trusted centre), to another entity. Key transport requires encryption of the key part, and this is typically done using asymmetric techniques and the public key of the counterpart. If used, for the establishment of secure messaging, key transport is used in both directions in order to derive a common key in the IFD and ICC.

Key agreement is the process of establishing a shared secret key between two entities A and B in such a way that neither of them can predetermine the value of the shared secret key. Key agreement mechanisms may be integrated in the device authentication.

In the context of session key establishment, 'implicit key authentication' means that after the execution of the mechanism only the participating entities can be in possession of the correct shared secret key.

This specification describes the following protocols:

- 3.5 Device authentication with privacy protection
- 3.6 Privacy constrained Modular EAC (mEAC) protocol with non-traceability feature
- 3.7 Symmetric authentication scheme
and the legacy protocol
- 3.8 Key transport protocol based on RSA

Information on the recommended use of the protocols can be found in Annex A.

3.6 Device authentication with privacy protection

3.6.1 General

In device authentication with privacy protection, to avoid the ICC disclosing private information, such as identity, a secure channel session is established before any other operation. To do so, the protocol starts with an DH:EC Diffie-Hellman key exchange and then authenticates the IFD first before the ICC. The notion “**DH**” indicates “unauthenticated Diffie-Hellman” whereas “**EC**” assigns “elliptic curve Diffie-Hellmann” – both are valid alternatives and the following steps reflect both of them equally.

The protocol is recommended wherever the identity of the ICC shall not be revealed to either the IFD, or to an attacker tapping the communication before the IFD has been authenticated.

Five main phases may be distinguished:

- i) Set the privacy protection.
- ii) Transport of the IFD public key to the ICC.
- iii) External authentication of the IFD.
- iv) Transport of the ICC public key to the IFD.
- v) Internal authentication of the ICC.

Usage of the privacy protocol mandates that the IFD is authenticated first.

3.6.2 Authentication steps

3.6.2.1 General

The following table shows device authentication with privacy protection and key agreement. The information in the fields is simplified. A detailed description of the coding is specified below.

DF.CIA shall only be readable to the extent that no personal information is read from the ICC until Step 11 is successfully accomplished. Until then, no information specific to the card holder (in contrast to the ICC's capabilities) shall be readable from DF.CIA.

Table 2 — Reference device authentication scheme 1 with Diffie-Hellman Key Exchange

Phase	Step	IFD	Transmission	ICC
1	1	READ BINARY of file DH DH: public parameters DH.P EC: public parameters ECDH.P.	→ ←	DH: Read DH key parameters p , q and g from specified file. EC: Read public parameters from specified file. Data are in response APDU
	1'(alt)	GET DATA for protocol related parameters DH: DH.P EC: ECDH.P		
	2	MSE:SET:AT ¹ Set [DH EC] DH algorithm	→ ←	OK

Phase	Step	IFD	Transmission	ICC
	3	DH: compute $K_{\text{IFD}} = g^{r_{\text{IFD}}} \bmod p$ EC: compute $K_{\text{IFD}} = r_{\text{IFD}} \cdot G$ GENERAL AUTHENTICATE(KID) ¹ (compliant to ISO/IEC 7816-4) DH: KID = DH.P K_{IFD} EC: KID = ECDH.P K_{IFD} Response = K_{ICC} DH: compute $K_{\text{IFD}/\text{ICC}} = K_{\text{ICC}}^{r_{\text{IFD}}} \bmod p$ EC: compute $K_{\text{IFD}/\text{ICC}} = r_{\text{IFD}} \cdot K_{\text{ICC}}$	→ ←	DH: compute $K_{\text{ICC}} = g^{r_{\text{ICC}}} \bmod p$ EC: compute $K_{\text{ICC}} = \text{Comp}(r_{\text{ICC}} \cdot G)$ DH: compute $K_{\text{IFD}/\text{ICC}} = K_{\text{IFD}}^{r_{\text{ICC}}} \bmod p$ EC: compute $K_{\text{IFD}/\text{ICC}} = \text{Comp}(r_{\text{ICC}} \cdot K_{\text{IFD}})$ K_{ICC}
Secure messaging keys are derived on both sides according to 3.9 and [15]. Secure messaging (encryption) is activated. All following commands are protected by secure messaging with encryption as described in “Final command APDU construction” [24] Chapter 9.6.3. The SSC is initialized to a start value of '1'.				
2	4	If PuK.CA.AUT is available in the ICC continue with step 7		
	5	Select PuK.RCA.AUT	→ ←	Select key. Response OK
	6	PSO:Verify Certificate C_CV.CA _{IFD} .CS_AUT	→ ←	Verify certificate. Response OK
	7	Select the CA public key PuK.CA _{IFD} .CS_AUT.	→ ←	Select key. Response OK
	8	PSO:Verify Certificate C_CV.IFD.AUT	→ ←	Verify certificate. Response OK

¹ For compliance to CWA 14890, the existence of the A6 template will indicate usage of the CWA 14890 method. This mechanism is restricted to the algorithms already supported in CWA 14890 in which case AES and elliptic curves are not supported.

Step 2' DH: compute $K_{\text{IFD}} = g^{r_{\text{IFD}}} \bmod p$

MSE:SET:KAT (A6)

DH: DH.P || K_{IFD}

Step 3' GET DATA (KICC)

DH: compute $K_{\text{IFD}/\text{ICC}} = K_{\text{ICC}}^{r_{\text{IFD}}} \bmod p$

Phase	Step	IFD	Transmission	ICC
The Public Key of the IFD is now known by the ICC, and can be trusted				
3	9	MSE:SET PuK.IFD.AUT Select IFD's public key and select the key parameters for the consecutive operation	→ ←	Select PuK of IFD Response OK
	10	GET CHALLENGE	→ ←	RND.ICC
	11	EXTERNAL AUTHENTICATE DH: SN.IFD DS[PrK.IFD.AUT] (... h(... K _{IFD} S)) with S = SN.IFD RND.ICC K _{ICC} DH.P EC: SN.IFD DS[PrK.IFD.AUT] (Comp(K _{IFD}) S) with S = SN.IFD RND.ICC Comp(K _{ICC}) ECDH.P	→ ←	ICC verifies signature. The IFD's identity is implicitly verified as known by the certificate C_CV.IFD.AUT. OK
ICC has now authenticated the reader — SM (encryption) active The full information of DF.CIA may now be revealed to the IFD				
4	12	READ BINARY of file EF.C.CA _{ICC} .CS_AUT ^b	→ ←	Read data from specified file. C.CA _{ICC} .CS_AUT as response
	13	READ BINARY of file EF.C.ICC.AUT	→ ←	Read data from specified file. C.ICC.AUT is in response APDU
The Public Key of ICC is now known by the reader, and can be trusted				
5	14	MSE:SET PrK.ICC.AUT	→ ←	Select PrK of ESIGN application. Response OK

Phase	Step	IFD	Transmission	ICC
	15	INTERNAL AUTHENTICATE (RND.IFD) IFD verifies signature. B's identity is implicitly verified as known from the certificate C.ICC.AUT	→ ←	The ESIGN application computes DH: $\text{SN.ICC} \parallel \text{DS}[\text{PrK.ICC.AUT}] \parallel \left(\dots \parallel h(\dots \parallel K_{\text{ICC}} \parallel S) \right)$ with $S = \text{SN.ICC} \parallel \text{RND.IFD} \parallel K_{\text{IFD}} \parallel \text{DH.P}$ EC: $\text{SN.ICC} \parallel \text{DS}[\text{PrK.ICC.AUT}] \parallel \left(\text{Comp}(K_{\text{ICC}}) \parallel S \right)$ with $S = \text{SN.ICC} \parallel \text{RND.IFD} \parallel \text{Comp}(K_{\text{IFD}}) \parallel \text{ECDH.P}$
Two-way authentication is now complete ^c				
a If the data are not organized in a file, it may be read from data objects instead. b This step is required only if the reader does not have the public key of the CA. c The SSC will be updated on both sides according to 8.10				

3.6.2.2 Step 1 — Read key exchange parameters

If the IFD does not have the required key exchange information available, it may read them from a global data file **DH:EF.DH** / **EC:EF.ELC** or data objects respectively. [EF.DH/EF.ELC](#) may contain public algorithm quantities which depend on the authentication algorithm. For instance:

- in a Diffie-Hellman Key exchange scheme, the public key quantities would be the public parameters g , p and q (refer to “Diffie-Hellman key exchange parameters” [24] Chapter 11.3).
- In an Elliptic Curve Diffie-Hellman Key exchange scheme, the public key quantities would be the public parameters p, a, b, G, n and the optional cofactor h (refer to “ELC key exchange public parameters” [24] Chapter 11.7).

Execution Flow

Table 3 — READ EF.DH parameters - command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'B0' READ BINARY
P1	'83' Example: SFI = 03
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.

Table 4 — Read EF.DH parameters — response

Response Parameter	Meaning
Data field	DH: data.EF.DH Data of EF.DH acc. to [24] Chapter 11.3 EC: data.EF.ELC Data of EF.ELC acc. to [24] Chapter 11.7
SW1-SW2	Refer to ISO/IEC 7816-4

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

Alternative Execution Flow with GET DATA

Table 5 — Get EF.DH parameters — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'CA' GET DATA
P1-P2	'7F49' DH key parameters
Le field	'00'

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.4.3.

DH: The contents of EF.DH are specified in chapter “EF.DH” [24] Chapter 13.5.

EC: The contents of EF.ELC are specified in chapter “EF.ELC” [24] Chapter 13.6.

Table 6 — Get EF.DH parameters — response

Response Parameter	Meaning
Data field	DH: data.EF.DH Data of EF.DH acc. to [24] Chapter 11.3. EC: data.EF.ELC Data of EF.ELC acc. to [24] Chapter 11.7.
SW1-SW2	Refer to ISO/IEC 7816-4

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

NOTE The public key quantities reveal information about the authentication mechanism. As long as these quantities are used in many ICCs, the identity of an ICC cannot be determined from this information.

3.6.2.3 Step 2 — IFD selects the public key parameter set and sends K_{IFD}

In order to establish a secure session, the IFD generates and sends it to the ICC within the CRT(AT) of an MSE command.

Execution Flow

Table 7 — Selection of key parameters — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'41' SET for key agreement
P2	'A4' CRT for AT
Lc field	$L_{A4} = L_{80} + L_{83} + 4$
Data field	'80' L_{80} [AlgID OID] '83' L_{83} < KeyRef >

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 are specified in 4.1.5.

The response data field is empty.

Table 8 — Selection of DH/ECDH key parameters — response

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

3.6.2.4 Step 3 — ICC computes K_{ICC}

NOTE According to ISO/IEC 7816-4 there are two ways to retrieve K_{ICC} .

3.6.2.4.1 IFD selects the DH/ECDH public key parameter set and sends K_{IFD}

K_{ICC} is returned in the response of a GET DATA command. see 3.5.1.3.3.

3.6.2.4.2 Use of GENERAL AUTHENTICATE APDU

K_{ICC} is returned in the response of a GENERAL AUTHENTICATE command.

Execution Flow

Table 9 — GENERAL AUTHENTICATE — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'86' GENERAL AUTHENTICATE
P1	'00'
P2	'00'
Lc field	$L(DO_{7C}) + L(DO_{7F49})$
Data field	refer to 4.1.6.
Le field	'00'

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.5.5.

The structure and coding of the key reference data are specified in 4.1.6.

The ICC generates:

$$\text{DH: } K_{ICC} = g^{r_{ICC}}$$

$$\text{EC: } K_{ICC} = \text{Comp}(r_{ICC} \bullet G)$$

and sends it to the IFD in the response of the command in the data field of a KAT template as DO.RND.

Table 10 — GENERAL AUTHENTICATE — response

Response Parameter	Meaning
Data field	'7C' L_{7C} '85' L_{85} K_{ICC}
	Optional transmission of the domain parameters can be part of the data (legacy) DH: '7F49' L_{7F49} DH Key Parm DOs (public DH parameters acc. to [24] Chapter 13.5) EC: '7F49' L_{7F49} ECDH Key Parm DOs (ECDH parameters acc. to [24] Chapter 13.6)
SW1-SW2	Refer to ISO/IEC 7816-4

The IFD (after having received the response of the GENERAL AUTHENTICATE command) and the ICC (during the execution and the context of the GET DATA command) establish a secure session using key derivation according to 3.9. The send sequence counter for MAC computation is set to the value of '1' for the next block sent by the IFD.

3.6.2.4.3 Retrieving K_{ICC} through GET DATA

This mechanism is applied for backwards compatibility only and therefore restricted to the algorithms already supported in CWA 14890.

According to CWA 14890-1 Steps 2' and 3' are subject to the following method. The existence of the A6 template will indicate usage of this method

IFD selects the DH public key parameter set and sends K_{IFD}

In order to establish a secure session, the IFD generates and sends it to the ICC within the CRT(KAT) of a MSE command.

Execution Flow

Step 2' — Selection of DH key parameters — command APDU

Command

Parameter Meaning

CLA according to ISO/IEC 7816-4

INS '22' MANAGE SECURITY ENVIRONMENT

P1 '41' SET for key agreement

P2 'A6' CRT for KAT

Lc 'xx'

Data '91' L_{91} PUK.IFD.KA ||

'84' L_{84} KID(PrK.ICC.KA) (conditional)

The KID is required if the private key is ambiguous, i.e. more than one key pair is available for Chip Authentication

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

The response data field is empty.

Selection of DH key parameters — response

Response

Parameter Meaning

Data field empty

SW1-SW2 Refer to ISO/IEC 7816-4

Step 3' — ICC computes K_{ICC}

The key portion K_{ICC} is returned in the response of a GET DATA command.

Execution Flow

Get Headerlist for KAT template — command APDU

Command

Parameter Meaning

CLA according to ISO/IEC 7816-4

INS 'CB' GET DATA

P1 '3F' current DF

P2 'FF'

Lc field 06'

Data field '4D 04 A6 02 91 00' // extended header list with KAT template

Le field '00'

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.4.4.

The ICC generates

$$\text{DH: } K_{\text{ICC}} = g^{r_{\text{ICC}}}$$

and sends it to the IFD in the response of the command in the data field of a KAT template as DO.RND.

Get Headerlist for KAT template — response

Response

Parameter Meaning

Data field 'A6' L_{A6} 91 L₉₁ K_{ICC}

SW1-SW2 Refer to ISO/IEC 7816-4

NOTE Setting the security environment in the first step implicitly establishes the correct handling of the transmitted random number. The KAT context, indicated in the extended header list of the GET DATA command selects the correct random number generation algorithm for K_{ICC}.

The IFD (after having received the response of the GET DATA command) and the ICC (during the execution and the context of the GET DATA command) establish a secure session using key derivation according to 3.9. The send sequence counter for MAC computation is set to the value of '1' for the next block sent by the IFD.

3.6.2.5 Step 4 — Skip reading chain certificates

If the public key PuK.CA_{IFD}.CS_AUT is already verified and trusted by the ICC, then steps 5 and 6 can be skipped.

3.6.2.6 Step 5 — Selection of verification key PuK.(R)CA_{IFD}.CS_AUT (conditional)

If PuK.CA_{IFD}.CS_AUT is not available in the ICC, a chain of (cross-certificates) shall be presented to the ICC, in order to confirm the validity of each certificate. The chain of certificates ends at any key which is available in the ICC. The top key of the chain, available in the ICC, is the public key of the root CA, named PuK.RCA.AUT. Steps 5 and 6 have to be performed repeatedly until the correct PuK.CA_{IFD}.CS_AUT is available in the ICC. This process is called multi-stage verification.

The MSE command selects the key to be used for verification of the certificate to be received in a consecutive command.

Execution Flow

Table 11 — Select CA root key — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'81' SET for verification
P2	'B6' DST (Digital Signature Template)
Lc field	'xx'
Data field	'83' L ₈₃ CRT(PuK.RCA.AUT < or other key in chain >)

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 are specified in 4.1.3.

The response data field is empty.

Table 12 — Select CA root key — response

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

3.6.2.7 Step 6 — Verify Certificate C_{CV}.CA_{IFD}.CS_{AUT} (conditional)

In multi-stage verification the certificate of a CA's public key C_{CV}.CA_{IFD}.CS_{AUT} is finally sent to the ICC with the VERIFY CERTIFICATE command. The ICC verifies this certificate with the public key set in the previous step. If the verification fails, the ICC responds with an appropriate status word. When successful, the ICC stores the PuK contained in the certificate and allows a further import of certificates along the certificate chain.

The role of the certificate holder shall comply with its signer. The selection of PuK.RCA can always be accepted to verify a chained certificate. Any other key, that was imported with a certificate may be applied only, if the role of its certificate matches the purpose of verification of the next incoming certificate.

Execution Flow

Table 13 — CA Certificate verification — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'2A' PERFORM SECURITY OPERATION
P1	'00' response data field absent
P2	'AE' VERIFY CERTIFICATE (RSA) 'BE' VERIFY CERTIFICATE (ELC)
Lc field	'xx'
Data field	C_CV.CA _{IFD} .CS-AUT CA's authentication certificate

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

RSA: The structure of the certificate is specified in 5.13.

ELC: The structure of the certificate is specified in 5.6.2.

The transmission of the certificate, however, is done without the tag '7F21' in compliance with the examples in ISO/IEC 7816-8:2004, A.3.

Only the relevant data objects from the value field of the certificate will be transmitted. These data objects may be transmitted with command chaining.

The response data field is empty.

Table 14 — CA Certificate verification — response

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

NOTE The execution flow of the VERIFY CERTIFICATE operation is outlined in 5.10.

According to this flowchart the ICC shall verify evidence that the role or the PuK of the IFD either corresponds to signing another (subsequently transmitted) certificate or to the authentication mechanism. The role (refer to 5.7.5.2) is part of the certificate when the PuK is imported. The role of the last PuK imported with VERIFY CERTIFICATE shall be stored in the ICC until the execution of the EXTERNAL AUTHENTICATE command in order to allow this verification.

3.6.2.8 Step 7 — Selection of verification key PuK.CA_{IFD}.AUT

The MSE command sets the keyID of the CA's public key to be used for the verification of the IFD's authentication certificate.

Execution Flow

Table 15 — Select IFD's CA key — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'81' SET for verification
P2	'B6' DST (Digital Signature Template)
Lc field	'xx'
Data field	['80' L ₈₀ AlgID] (conditional) ¹ '83' L ₈₃ CRT(PuK.CA _{IFD} .CS_AUT < or other key in chain >)
¹ The algorithm ID may be omitted if already known through previous steps	

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 are specified in 4.1.3.

The response data field is empty.

Table 16 — Select IFD's CA key — response

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

3.6.2.9 Step 8 — Verify Certificate C_CV.IFD.AUT

The IFD sends a card verifiable (CV) certificate C_CV.IFD.AUT along with the VERIFY CERTIFICATE command. The ICC verifies the certificate using the public key PuK.CA_{IFD}.CS_AUT of the certification authority CA.

The certificate contains the IFD's PuK.IFD.AUT, which is stored temporarily by the ICC.

Execution Flow

Table 17 — IFD Certificate verification — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'2A' PERFORM SECURITY OPERATION
P1	'00' response data field absent
P2	'AE' VERIFY CERTIFICATE (DH) 'BE' VERIFY CERTIFICATE (ELC)
Lc field	'xx'
Data field	C_CV.IFD.AUT IFD's authentication certificate

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

RSA: The structure and coding of the card verifiable certificate is specified in 5.12.

ELC: The structure of the certificate is specified in 5.6.2.

The response data field is empty.

Table 18 — IFD Certificate verification — response

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

NOTE The execution flow of the VERIFY CERTIFICATE operation is outlined in 5.10.

According to this flowchart the ICC shall verify evidence that the role for the PuK of the IFD either corresponds to signing another (subsequently transmitted) certificate or to the authentication mechanism. The role (refer to 5.7.4) is part of the certificate when the PuK is imported. The role of the last PuK imported with VERIFY CERTIFICATE shall be stored in the ICC until the execution of the EXTERNAL AUTHENTICATE command in order to allow this verification.

After Step 8 the IFD's public authentication key PuK.IFD.AUT can be trusted.

3.6.2.10 Step 9 — Key Selection for external authentication

External authentication involves a challenge from the ICC to be signed with the secret authentication key (PrK.IFD.AUT) of the IFD. The appropriate verification of this signature in the ICC requires the appropriate public authentication key (PuK.IFD.AUT).

In case of existence of more than one key in the ICC usable for external authentication the key to be used shall be selected in advance. This selection is performed with the MANAGE SECURITY ENVIRONMENT (MSE) command.

Execution Flow

Table 19 — Set IFD's authentication key — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'81' SET for external authentication
P2	'A4' AT (Authentication Template)
Lc field	'xx'
Data field	'80' L ₈₀ AlgID '83' L ₈₃ < KeyRef of PuK.IFD.AUT >

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

For the structure and coding of the data field refer to 4.1.4.

The response data field is empty.

Table 20 — Set IFD's authentication key — response

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

3.6.2.11 Step 10 — 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.

NOTE Cryptologically K_{IFD} and K_{ICC} computed in Step 2 and 3 have sufficient random quality. The additional request for RND.ICC is used to initialize the SSC counters (refer to “Send sequence counter (SSC)” [24] Chapter 9.5)

Execution Flow

Table 21 — GET CHALLENGE — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 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 22 — GET CHALLENGE — response

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

3.6.2.12 Step 11 — External authentication

The IFD computes a signature on the concatenation of the challenge with its own key token information K_{IFD} using PrK.IFD.AUT .

After the ICC verifies the signature with the trusted public key PuK.IFD.AUT , it responds with a confirmation whether the authentication was successful or not.

Execution Flow

Table 23 — 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	DH: SN.IFD SIG with $SIG = DS^a[PrK.IFD.AUT]('6A' PRND1 h[PRND1 K_{IFD} SN.IFD RND.ICC K_{ICC} DH.P] 'BC')$ EC: SN.IFD SIG with $SIG = DS^b[PrK.IFD.AUT] (Comp(K_{IFD}) SN.IFD RND.ICC Comp(K_{ICC}) ECDH.P)$
a DS is to be understood as plain RSA exponentiation. It does not imply a layer of input formatting other than already specified here. b DS is to be understood as a plain EC-DSA based signature. It does not imply a layer of input formatting other than already specified here.	

For the structure and coding of the data field refer to 4.3.1 and 4.3.2.

For the structure and content of the APDU refer to ISO/IEC 7816-4:2013, 11.5.4. For PRND1 refer to Table 127.

The public key compression function *Comp(...)* is defined [24] Chapter 11.5

Table 24 — EXTERNAL AUTHENTICATE — response

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

The ICC shall verify evidence, that the role for the PuK of the IFD corresponds to the external authentication mechanism. The role (refer to 5.7.4) is part of the certificate when the PuK was imported. The role shall be stored in the ICC temporarily between the execution of the VERIFY CERTIFICATE command and the EXTERNAL AUTHENTICATE command in order to allow this verification.

3.6.2.13 Step 12 — Read C.CA_{ICC}.AUT (conditional)

The IFD obtains PuK.CA_{ICC}.CS_AUT by reading C.CA_{ICC}.CS_AUT from a file EF.C.CA_{ICC}.AUT in the ICC. Then it verifies the certificate C.CA_{ICC}.CS_AUT and extracts and stores the public key PuK.CA_{ICC}.CS_AUT temporarily for use in step 13.

Execution Flow

Table 25 — Read C.CA_{ICC}.AUT certificate — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'B0' READ BINARY
P1	'83' Example: SFI = 03
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 5.13, or as Self-descriptive certificates according to 5.8.2 but it could also be some other format unknown to the ESIGN application. The actual format is application specific and out of the scope of this standard.

The response data contains the ICC's CA certificate.

Table 26 — Read C.CA_{ICC}.AUT certificate — response

Response Parameter	Meaning
Data field	data.EF.C.CA _{ICC} .CS_AUT
SW1-SW2	Refer to ISO/IEC 7816-4

3.6.2.14 Step 13 — Read ICC's certificate C.ICC.AUT

The IFD reads the ICC's certificate to be used for later authentication.

The received certificate C.ICC.AUT contains the ICC's public key. The IFD verifies the certificate using the public key of the certification authority PuK.CA_{ICC}.CS_AUT. If C.ICC.AUT can be verified successfully, the PuK.ICC.AUT is stored in the IFD and can be trusted for subsequent authentication.

Execution Flow

Table 27 — Read C.ICC.AUT certificate — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'B0' READ BINARY
P1	'84' Example: SFI = 04
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.

Refer to 5.14 for more information concerning the structure and content of this certificate. The response data contains the ICC's certificate.

Table 28 — Read C.ICC.AUT certificate — response

Response Parameter	Meaning
Data field	data.EF.C.ICC.AUT
SW1-SW2	Refer to ISO/IEC 7816-4

3.6.2.15 Step 14 — Key selection

Before processing the INTERNAL AUTHENTICATE command, the ICC's private authentication key PrK.ICC.AUT shall be set by the MSE command.

The MSE command updates the current security environment.

NOTE In general, the availability and usability of the selected key(s) need to be checked. It is not the role of this standard to mandate in which step these checks occur.

Execution Flow

Table 29 — Select ICC authentication key — 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' AT (Authentication Template)
Lc field	'xx'
Data field	'80' L ₈₀ AlgID '84' L ₈₃ < KeyRef of PrK.IFD.AUT > value field of "CRT for selection of ICC's PrK.ICC.AUT"

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

For the coding of the key reference data refer to 4.1.1.

The response data field is empty.

Table 30 — Select ICC authentication key — response

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

3.6.2.16 Step 15 — Internal Authentication

The IFD performs an INTERNAL AUTHENTICATE command. The challenge sent to the ICC with this command is RND.IFD.

The ICC then computes the signature over the challenge and the key tokens K_{IFD} , K_{ICC} and returns it to the IFD encrypted with secure messaging.

The IFD verifies the response with the trusted key of the ICC: PuK.ICC.AUT and obtains evidence that the signer (holder of the certificate) is identical with the entity that made the key negotiation.

Execution Flow

Table 31 — INTERNAL AUTHENTICATE — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'88' INTERNAL AUTHENTICATE
P1	'00'
P2	'00' key reference — none
Lc field	'xx'
Data field	RND.IFD
Le field	'00'

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

For the content of the response data field refer to 4.3.3 and 4.3.4.

The signature is formatted according to ISO/IEC 9796-2 (DS scheme 1).

The response data contains the signature.

Table 32 — INTERNAL AUTHENTICATE — response

Response Parameter	Meaning
Data field	DH: DS = SN.ICC S with S = DS ^a [PrK.ICC.AUT] ('6A' PRND2 h(PRND2 K_{ICC} SN.ICC RND.IFD K_{IFD} DH.P) 'BC') EC: DS = SN.ICC S with S = DS ^b [PrK.ICC.AUT] (Comp(K_{ICC}) SN.ICC RND.IFD Comp(K_{IFD}) ECDH.P)
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. b DS is to be understood as an ECDSA. It does not imply a layer of input formatting other than already specified here.	

For PRND2 refer to Table 128.

NOTE Cryptologically K_{IFD} and K_{ICC} computed in Steps 2 and 3 have sufficient random quality to provide freshness. RND.IFD is not cryptologically required. Using RND.IFD = '00 .. 00' adds non-traceability feature to the protocol.

3.7 Privacy constrained Modular EAC (mEAC) protocol with non-traceability feature

3.7.1 General

The protocol presented in this chapter is to establish a trusted channel between the ICC and a server (possibly remote) via an IFD. The protocol also allows the establishment of a secured channel between the server and the ICC (i.e. a channel with established secure messaging (SM) between the end-to-end devices) as well as user verification according to chapter [24] Chapter 6 which can be executed explicitly² or implicitly³.

The mEAC protocol is separated in two parts: Terminal Device Authentication (TDA) allows the authentication of the IFD or remote server, whereas Chip Device Authentication (CDA) proves identity of the ICC. Three members of the mEAC protocol family are distinguished in this standard:

- “mEAC non-traceability” as described in the previous version of this standard (2008) refers to the sequences TDA-CDA
- mEACv1 refers to the sequence CDA-TDA in EACv1.1 [13].
- mEACv2 refers to the sequence TDA-CDA in EACv2.0 [14].

Only mEACv1 and mEACv2 will be presented in this standard.

The CDA protocol provides for the special case of non-traceability due to the following property:

The protocol prevents a verifier submitting a provable existence of the verification process. A possible application is explained in the following example. The protocol is popularly being used with contactless implementations in the context of citizen cards and homeland security. It replaces the Active Authentication protocol (refer to [16]) by aspects of privacy.

3.7.2 Example for traceability case

When Alice crosses the border of “Dangerland” her ICC is authenticated. If the signature that Alice provides to the officer of “Dangerland” contains provable evidence that Alice visited “Dangerland”, the officer could send a notice to “Homeland” that Alice was there.

When Alice returns to her “Homeland” she might be sued because “Homeland” has proven evidence (by Alice's signature). Of course “Dangerland” can always report Alice's evidence, however, if non-traceability is given “Homeland” does not have a mathematical proof and therefore cannot sue Alice due to lack of evidence.

² For example using a VERIFY PIN command

³ For example using an authentication protocol according to this standard, e.g. PACEv2, see ISO/IEC FDIS 11770-4:2005 . Information technology – Security techniques – Key Management – Part 4: Mechanisms based on weak secrets, 2005.

3.7.3 Notation

The following notation is used.

AUX.Data	Auxiliary Data (see explanation below)
PrK.IFD.AUT	IFD's private key for Authentication
PuK.IFD.AUT	IFD's public key Authentication
PrK.ICC.KA	ICC's private key for Key agreement
PuK.ICC.KA	ICC's public key for Key agreement
PrK.IFD.KA	IFD's private key for Key agreement
PuK.IFD.KA	IFD's public key for Key agreement
PuK.RCA.AUT	Public Key of Root CA stored in ICC for authentication purposes
C_CV.CA_{IFD}.CS_AUT	Card verifiable certificate issued by the IFD root CA or underlying CAs used to authenticate the IFD public key
PuK.CA_{IFD}.CS_AUT	Public key contained in certificate C_CV.CA_{IFD}.CS_AUT for authentication purpose
C_CV.IFD.AUT	card verifiable certificate that contains the IFDs public key used in authentication procedure
C.ICC.AUT	certificate for ICC PuK.ICC.KA
RND.ICC	random number generated by ICC
RND2.ICC	2nd random number generated by ICC
K_{ENC}, K_{MAC}	Session Keys for encryption and MAC-calculation

Auxiliary Data: The asymmetric protocols allow the IFD to send optional auxiliary data. The purpose of this data are not under the scope of this standard. A typical use case may be the age verification threshold to be sent as AUX.Data object to the ICC. If the ICC compares the received threshold value to an internally stored age, it might deny the following service, e.g. cigarette vending to under-aged persons.

The COMPARE command is used in Part 2 for the comparison of Auxiliary data, e.g. AUX DATA, as optionally transmitted in this authentication protocol. The compare functionality is described in [27] Chapter 3.2.

3.7.4 Authentication steps

3.7.4.1 General

The protocols mEACv1 and mEACv2 differ in the order of performing protocols TDA and CDA. The different orders and content of protocol steps are given in Table 33 and Table 34. In mEACv2, TDA (see steps 3 to 9 in Table 34) is performed before CDA (see steps 10 to 13 in Table 34) whereas in mEACv1 CDA is performed before TDA.

Hence, in mEACv1 the protocol steps 10 to 13 in Table 34 shall be performed prior to steps 3 to 9.

Table 33 — Step execution order in mEACv1 versus mEACv2

Order	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
mEACv1	1	2	10	11	12	13	3	4	5	6	7	8	9				
mEACv2	1	2					3	4	5	6	7	8	9	10	11	12	13

Table 34 — Reference device authentication scheme mEAC

Stage	Step	IFD	Transmission	ICC
A	1	READ BINARY of file containing protocol related parameters (e.g. public parameters etc.)	→	Read parameters Data are in response APDU
		GET DATA to read public parameters from EF.DH.	←	
Protocol relevant data are available to the terminal.				
B	2 conditional	In case of a remote interface: Proof of user presence: User is requested to prove presence by presentation of e.g. PIN	→ ←	Verify user presence Response: OK
In case of a remote interface or a contactless local interface, proof of user presence shall be given by mechanisms defined in “User verification” [24] Chapter 6 in order to avoid skimming attacks.				
	10	For mEACv1 perform the steps 10–13 here. For backward compatibility to EACv1.1 protocol variant, step 11 and step 12 may alternatively be replaced by step 11’ applying MSE SET KAT command (see 3.6.3.11)		
	11			
	12			
	13			
	3	Select and/or verify PuK.RCA.AUT.	→ ←	Select and/or verify key. Response: OK
	4	PSO:Verify Certificate C_CV.CA _{IFD} .CS_AUT	→ ←	Verify certificate.
	The verification of certificates along a certificate chain may continue until the CA's key is available in the ICC that signed the IFD's certificate and delivers PuK.IFD.AUT .			
	5	Select and/or verify PuK.CA _{IFD} .CS_AUT.	→ ←	Select and/or verify key. Response: OK
	6	PSO:Verify Certificate C_CV.IFD.AUT	→ ←	Verify certificate. Response: OK

The Public Key of the IFD is now known to the ICC, and can be trusted.				
D	7	MSE:SET AT (PuK.IFD.AUT) Select IFDs public key and select the key parameters for the consecutive operation, send compressed ephemeral public key $Comp(PuK.IFD.KA)$ and optional auxiliary data For the use of AUX.Data refer to [27] Chapter 3.2	→ ←	Select PuK of IFD, store $Comp(PuK.IFD.KA)$ store auxiliary data (optional) Response: OK
	8	GET CHALLENGE	→ ←	RND.ICC
	9	EXTERNAL AUTHENTICATE By computing the digital signature $DS[PrK.IFD.AUT](S)$ with $S = [SN.ICC \parallel RND.ICC \parallel Comp(PuK.IFD.KA) \parallel AUX.Data]$	→ ←	ICC verifies signature. The IFD's identity is implicitly verified as known by the certificate C_CV.IFD.AUT. Response: OK
ICC has now authenticated the IFD				
E	10	Conditional – only if C.ICC.AUT is available: READ BINARY to obtain C.ICC.AUT or PuK.ICC.KA	→ ←	Read C.ICC.AUT or PuK.ICC.KA Data are in response APDU
	11	MSE SET AT Set PrK.ICC.KA for authentication	→ ←	Set PrK.ICC.KA Response: OK
	12	GENERAL AUTHENTICATE (compliant to ISO/IEC 7816-4) send PuK.IFD.KA Generate new session keys K_{ENC} and K_{MAC} Compute $MAC[K_{MAC}](PuK.IFD.KA)$ and compare with received value	→ ←	Compute $Comp(PuK.IFD.KA)$ and compare with hash value obtained in step 7. Generate new session keys K_{ENC} and K_{MAC} . Response: OK and $MAC[K_{MAC}](PuK.IFD.KA)$ and RND2.ICC
	13	Establish secure session The secure messaging session is restarted with session keys generated in step 11 or 12.		
Secure channel (ICC- server) is now established and two-way authentication is complete; If Step 10 is not performed, then Step 10 shall be performed in secure messaging mode as Step 14 (not shown in this example).				

3.7.4.2 Step 1 — Reading of the protocol relevant public parameters

The IFD reads the protocol relevant public parameters to be used. The public parameters may be provided in the Cryptographic Information Application, in a CardInfo file [17] outside the ICC or in an elementary file EF.CardAccess defined in [14]. If an elementary file is used, a READ BINARY command is applied to retrieve the parameters.

Execution Flow

Table 35 — Read public parameters — command APDU

Command Parameter	Meaning
CLA	according to ISO/IEC 7816-4
INS	'B0' READ BINARY
P1	'84' Example: SFI = 04
P2	'00' offset = 0
Le field	'xx' number of bytes to read

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

Table 36 — Read public parameters — response

Response Parameter	Meaning
Data field	DH: Diffie-Hellman parameters according [24] Chapter 13.5 EC: Diffie-Hellman parameters according [24] Chapter 13.6
SW1-SW2	Refer to ISO/IEC 7816-4

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

NOTE The public key quantities reveal information about the authentication mechanism. As long as these quantities are used in many ICCs, the identity of an ICC cannot be determined from this information.

3.7.4.3 Step 2 — User verification (conditional)

At this stage, in case of a remote interface, user verification (acc. to chapter [24] Chapter 6) is possible. This can be performed implicitly (e.g. using a password-based protocol with the establishing of session keys for Secure Messaging– not standardized here) to protect additionally the retrieval of the PuK.ICC contained in ICCs certificate by Secure Messaging in step 10 (single step).

3.7.4.4 Step 3 — Selection of verification key PuK.(R)CA_{IFD}.CS_AUT (conditional)

If PuK.CA_{IFD}.CS_AUT is not available in the ICC a chain of (cross/link-certificates) shall be presented to the ICC in order to confirm the validity of each certificate. The chain of certificates starts at any key which is available in the ICC. The top key of the chain available in the ICC is the public key of the root CA, named PuK.RCA.AUT. Steps 5 and 6 in Table 34 have to be performed repeatedly until the correct PuK.CA_{IFD}.CS_AUT is available in the ICC. This process is called multi-stage verification.

The MSE command selects the key to be used for verification of the certificate to be received in a consecutive command.