



TECHNICAL REPORT

restricted identification through
**Access to e-Services with privacy-
preserving credentials**

Table of contents

Document history	3
Scope	4
Accessing a service with privacy-preserving credentials	4
Respective role of stakeholders.....	4
Handling of profiles and privacy-preserving credentials.....	5
Description of main steps	5
Example use for pseudonyms.....	10
Registration under a pseudonym.....	10
modular Enhanced Role Authentication (mERA).....	10
Sequence diagram	11
Authentication protocol description.....	12
Notation.....	12
Authentication flow.....	14
Stage 1 – Set the cryptographic context.....	17
Stage 2 – Generation of the authentication keys.....	17
Stage 3 – External authentication of the IFD (C1)	18
Stage 3 – Internal authentication of the ICC (C2)	20
Stage 4 – Certificate verification	21
Stage 5 – Retrieval of Public parameters for Key agreement.....	22
Stage 6 –Key agreement	23
Cryptographic suites	27
Certificate format	28

Document history

Contributors	Date	Main Changes	Comments
Gemalto Patent proposal from O.Joffray, A.Gouget, M.Faher, G.Tripotin	June 22nd 2010	Published with no IP rights	Proposed as contrib to ECC & EN14890
Mourad Faher –Gemalto	July 15th 2010	Issuance of first version	Rev.1 & 2
M.Faher, F. Peticara, G.Tripotin, B.Peirani-Gemalto, Sebastien Gelgon- ANTS, A.Feraud-Oberthur.	July 22nd 2010	Rev.2	overall review & approval of the draft topics and clauses
Gixel Members	Aug 18th 2010		Review & comments
M.Faher & Alban Feraud –Oberthur.	Aug. 23th 2010	Rev. from 3 to 6	Continuing review
A.Namour - Morpho	Aug. 24th 2010	Rev.6	Comments
M.Faher & Alban Feraud –Oberthur .	Aug. 25th 2010	Rev. 7	Editorial comments
M.Faher & Beatrice Peirani -Gemalto	Aug. 25th 2010	Rev. 8	Review and approval of commands set
M.Faher & Beatrice Peirani & A.Gouget –Gemalto, A.Feraud – Oberthur	Sept 2010	Continuing review of Rev.8	Review and approval of commands set, protocol security strengthening
M.Faher & Beatrice Peirani & A.Gouget –Gemalto, A.Feraud & Y.Sierra – Oberthur, S.Gelgon - ANTS	Oct 5th 2010	Rev.8	review of security measures ensuring tamperproof protocol
M.Faher & Beatrice Peirani & A.Gouget –Gemalto	Oct 14th 2010	Rev.9	protocol security strengthening with certificate verification
M.Faher & Beatrice Peirani & A.Gouget –Gemalto, A.Feraud & Y.Sierra – Oberthur, S.Gelgon - ANTS	Oct 18th 2010	Rev.9	Continuing review
M.Faher & Beatrice Peirani & A.Gouget –Gemalto, A.Feraud – Oberthur	Oct 25th 2010	Rev.9	Continuing review
M.Faher & F.Peticara & Gemalto R&D (O.Joffray, J.-P.Truong)	Oct 26th 2010	Rev.9	Tech. details review and action points for pending semantic definition
M.Faher & Beatrice Peirani & A.Gouget –Gemalto, A.Feraud – Oberthur	Oct 28th 2010	Rev.9	Details review Finalization of command sets and Ed. corrections. examination of WG16 comments
M.Faher (& Gemalto R&D)	Oct 29th 2010	Rev.10	Issuance of Rev.10 including the complete auth. flow acc. former resolutions and correction of pending mistypo + text justif.
M.Faher	Nov 19th 2010	Rev.11	Improvement of sequence diagram (same mechanism but better presentation as agreed in last WG15)
M.Faher	Dec 1st 2010	Rev. 11	update of fig-1 for clarification
M.Faher – Gemalto & A. Namour - Morpho	Dec 20th 2010	Version 1.rev11	general review of the doc
B.Peirani - Gemalto	Feb 1st 2011	Version 1.rev12	Cryptosuites corrected in table 19
M.Faher	April 30th 2011	Version 1.2	Table 21 completed with CHA field
M.Faher, A.Gouget, B.Peirani	June 28th 2011	Version 1.3	Update of fig.3 further to WG16 of Starnberg
M.Faher	Aug 10th 2011	Version 2.0	Addition of expiration date for the Criteria List + Ed. updates, removal of draft use case.

Scope

This document is a Technical Report. Its goal is to describe a protocol for access to e-Services with privacy-preserving credentials. It finds a model that aims at achieving user access to e-Services while preserving user private data from disclosure to the Service provider and preventing the Identity Provider from acquiring knowledge about the very nature of the service(s) requested by the user from the Service Provider. That is, the Service Provider is provided the proof that the cardholder is properly authenticated and fulfils the criteria list set down by the Service Provider. This proof is supplied by a trusted third party, here called "Identity Provider" how is not forcibly the Issuer, but how may be mandated by the Issuer.

The present report served as provision for contribution to CEN/TC224/WG15 and WG16.

Accessing a service with privacy-preserving credentials

The approach proposed herein solves the challenge of accessing a restricted e-service that requires a given user profile ⁽¹⁾ in the following conditions :

- without disclosing user's private data to the Service Provider
- proving implicitly to the Service Provider that the user fulfils the profile access criteria
- preventing IdP from acquiring knowledge about the very nature of the service requested by the card bearer to the Service Provider

According this paradigm, the card holder who wants to access a service may have to prove that his profile fulfils a set of criteria (e.g. age, group or club membership, community belonging, status, nationality, etc...) depending on Service Provider's requirements/policy. The proposed solution permits a card to get a set of privacy-preserving credentials ⁽²⁾ serving as a runtime-calculated pseudo-certificate from an identity provider.

NOTE 1 : List of criteria submitted by a Service provider to the user. The answers are needed by the service provider in order to grant access or not to the services

NOTE 2 : list of answers obtained by the IdP to the questions submitted by the service provider (SP). These answers together with the public part or the ephemeral key pair are electronically sealed by the IdP : The IdP protects the integrity of these data by signing them

Respective role of stakeholders

- Service Provider :

delivering upon request the criteria list (profile) conditioning access to a service; granting access to the service if the proof of criteria fulfillment is presented; setting the criteria list such as not to disclose to the Identity Provider the nature of services that are provided.

- Identity Provider :

verifying upon request whether the criteria list is fulfilled through questions asked to the card ; generating credentials that denotes for each of the criterium whether it is

fulfilled or not; applying a digital signature on the credentials and on an ephemeral public key delivered by the card.

- user (User Agent) :

requesting the profile for a given service; validating this profile on behalf of an Identity Provider; presenting the privacy-preserving credentials to the Service Provider.

- Card Issuer :

assigning a daughter secret key to each Service Provider so that this key can be generated by the card from a Master key personalized on-card.

Handling of profiles and privacy-preserving credentials

To achieve the protocol, three sets of data are conveyed through the card between the IdP and the SP and require appropriate storage and handling :

- the profile (*criteria list*) delivered by the SP
- the *privacy-preserving credentials* delivered by the IdP
- the *public part of the ephemeral key pair* generated on-card

Either dedicated Elementary files or constructed Data Objects may be employed to store these data. In the present embodiment of this protocol, constructed Data Objects are employed. Accordingly, 3 tags are assigned to encapsulate these data structures.

The structure of the data is to be determined for interoperability e.g BER-TLV, XML etc In the present embodiment of this protocol, BER-TLV encoding is employed. An appropriate syntax and semantic may be determined to leverage interoperability. CIA / EF.DCOD may be used to achieve interoperability for the tag allocation to the three sets of named data. Once Applicative tags are assigned to these data sets (i.e 7816-6), they should preferably be used in the future.

To ensure that *privacy-preserving credentials* values are recognized by their respective service provider to whom they are intended, the following mechanism is employed: the Data Objects containing these values are protected with access rules requiring a Role Authentication. This is, the Service Provider authenticates to the card to prove his ownership of a secret key. Consequently, *privacy-preserving credentials* supplied by Identity Providers can only be read by intended recipients, thereby preventing disclosing user data to unauthorized Service Providers.

Description of main steps

1) the user gets connected to a service and retrieves on his card the conditional access criteria required by the Service Provider. The user checks whether the service provider requirements (criteria) are acceptable

2) prior to the storage of the profile on-card, the user authentication via PIN presentation is required (user consent)¹, then the Service Provider authenticates to the card to prove his role (authorization to store profiles on-card)².

3) the profile is stored securely onto the card. Along with the profile, the Service Provider may optionally supply a profile identifier (a Pseudo³ belonging to the SP and not disclosed to the IdP) if needed to link the profile to some resource (i.e back-office, DB, user account, etc). Then the card generates an ephemeral key pair serving to securely connect later to this Service Provider.

4) afterwards, the user gets to an Identity Provider amongst a list of IdP entitled to deliver privacy-preserving credentials. A Mutual Authentication takes place between the IdP and the card : the IdP authenticates first to the card. A Secure Channel is established between the card and the IdP. The protocol employed for this purpose depends on security policy.

5) once a secure channel is established between the card and the IdP, the IdP may check whether the card is not in a revocation list; The IdP may check whether the card needs to be updated and may update it accordingly or just make the card bearer aware of this.

6) the IdP recovers both the public part of the ephemeral key pair formerly generated by the card (called PuK.KA.ICC) or its hash code, and the conditional access criteria (profile). The IdP informs accordingly the user of the very nature of these criteria, including their respective criterion requirement (indicating either "Mandatory" or "Optional"), thereby asking for user's consent : the user can reject for instance to disclose his age or his community-identifier or his specific-group-belonging etc, that he may consider not relevant for the service. Before to deliver the credentials, the IdP makes the necessary verifications (command COMPARE acc. ISO/IEC 7816-4) e.g. to check whether the card bearer is over 18 years old; optionally the IdP may provide the card with the certified current date.

7) the IdP applies his signature on all the credentials delivered to the card whereby proving their validity to the Service Provider; the credentials may optionally be timely restricted with a validity deadline (i.e access to the service may be restricted to a determined period : the operation of granting the card bearer rights to access restricted service can either be timely correlated or uncorrelated with the actual access to the service; in the former case, the right is granted for a determined duration, and in the latter case, the right is granted without expiration date. In case of expiration date, the SP may incorporate in the criteria list a Data Object of Tag 0x18 denoting a GeneralizedTime⁴ information. If the IdP notice any misfit between the Criteria with the expiration date

¹ a password based mechanism may be employed to prevent eavesdropping on the local segment of the communication line. This feature applies to both steps (2) and (4) of the sequence.

² modular Enhanced Role Authentication (mERA) is performed.

³ i.e. a 16-byte length Pseudo can be used

⁴ GeneralizedTime shall be encoded as [UNIVERSAL 24] Type modeling a date and an hour by means of a character string conforming to ISO 8601.

Figure 2 below shows the general view of the protocol through functional main steps.

Figure 3 below shows a sequence diagram of the protocol with numbered main steps according the description above

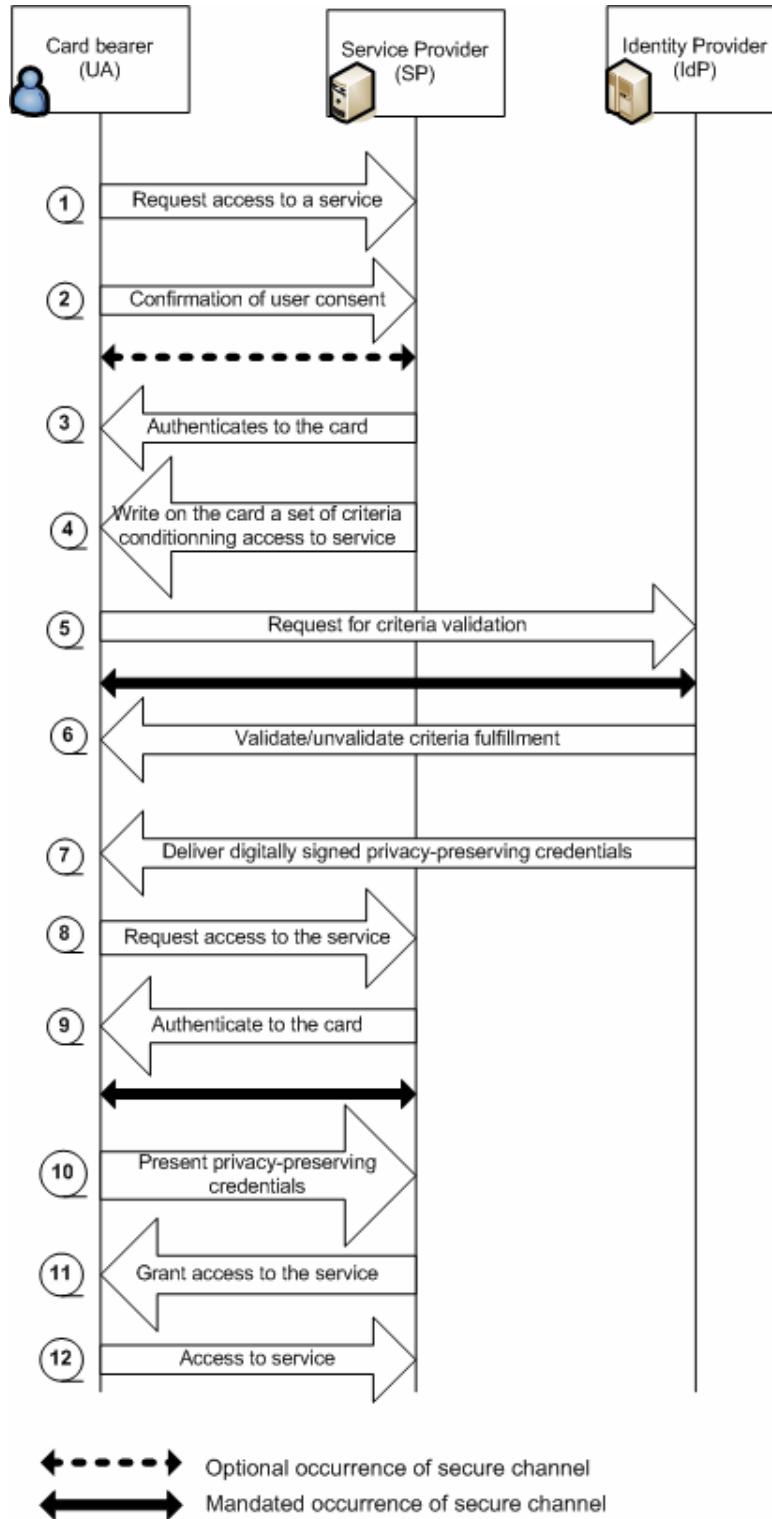


Figure 2 — Protocol overview : main functional steps

Example use for pseudonyms

Amongst the use cases supported by the *privacy-preserving credentials* alternative, the pseudonyms may be handled as follows (list non exhaustive).

Registration under a pseudonym :

- the card holder fills out a eService HTML form with a pseudonym instead of his real name and the eService turns the form field into a criterion such as "*is card holder's real name equivalent to the pseudonym ?*" This criteria will be revoked by the IdP during the process of *privacy-preserving credentials* delivery, and when the card holder gets back to the eService, the SP will guess through the credentials that the user filled out the registration form with a pseudonym that does not match his actual name.

- a pseudonym personalized on the card as part of user's privacy attributes may be used by the card holder and questioned by the eService through the criterion : "*is a card holder's pseudonym equivalent to the pseudonym registered with the service?*"

modular Enhanced Role Authentication (mERA)

In the running course of the protocol herein described, the Service Provider authenticates twice to the card :

- before the SP is authorized to write the set of criteria onto the card,
- before the SP is authorized to read the privacy-preserving credentials from the card

Accordingly, the Role Authentication protocol performed at steps (3) and (9) of Figure 2 or at steps (2) and (9) of Figure 3 is a cryptographic modular mechanism, the mERA (modular Enhanced Role Authentication) ref. 14890-2.

mERA details and APDU commands description are provided in clause " Authentication protocol description" below.

Sequence diagram

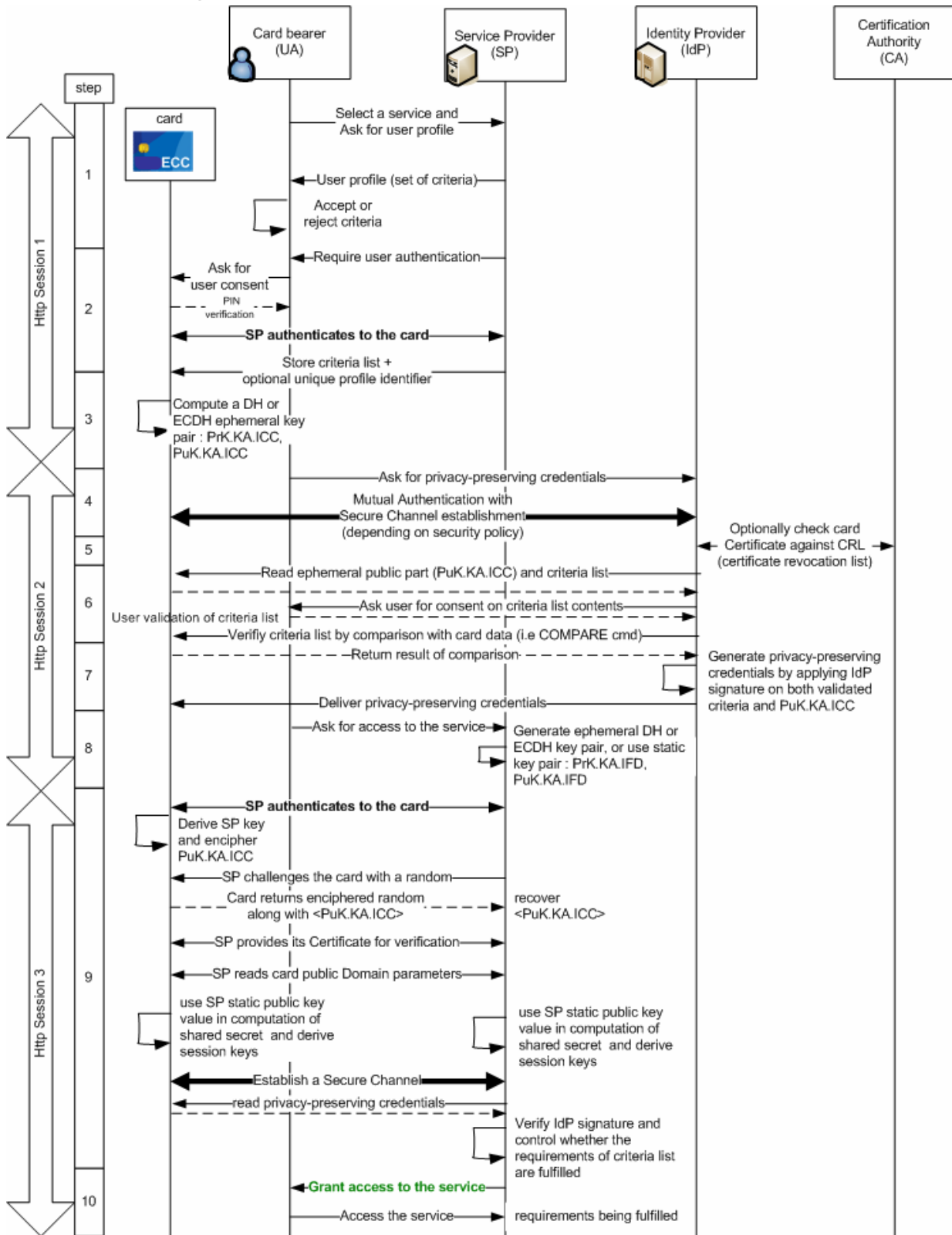


Figure 3 — Access to e-Service with privacy-preserving credentials

Authentication protocol description

Notation

Application-specific payload	Data specific to the application (e.g. public key, criteria list or Auxiliary data...), may comprise additionally a descriptor;
MK.ICC	Master secret key for authentication owned by the ICC;
SK.IFD	Authentication secret key of the IFD;
SN.IFD	Serial number of the IFD – It used to build SK.IFD from MK.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 the mERA to use;
ENC	Encryption algorithm associated to the flavour of mERA;
MAC	Checksum computation algorithm associated to the flavour of mERA;
KDF	Key Derivation function associated to the flavour of mERA;
KA	Key Agreement (may be DH or ECDH);
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.KA.IFD,PrK.KA.IFD	ephemeral key pair for key agreement of the IFD;
PuK.KA.ICC,PrK.KA.ICC	ephemeral key pair for key agreement of the ICC;
PuK.GENUINE.IFD,PrK.GENUINE.IFD	static asymmetric key pair for genuineness of the IFD;
D	Domain parameters used by the ICC. They match the asymmetric key pairs (PuK.KA.ICC,PrK.KA.IFD) & (PuK.KA.ICC,PrK.KA.ICC). DH or ECDH may be used;
CA	Certification authority known by the ICC that ensures genuineness of the IFD and certifies (signs) PuK.GENUINE.IFD;
CAR	Reference of the certification Authority CA;
CV	Card Verifiable Certificate(s) the CA issued for the IFD certifying PuK.GENUINE.IFD – This certificate(s) is verified by the ICC and contains PuK.GENUINE.IFD. The format of this certificate(s) is detailed in section “Certificate format”;

Authentication flow

stage	step	IFD	Trans- mission	ICC
1	A	Generate Random RND1.IFD Select the application-specific payload to retrieve MSE SET AT(AlgID,MK.ICC ref., SN.IFD RND1.IFD) ⁶	→ ←	Derive IFD secret key from MK.ICC (Master Key) $SK.IFD = KDF [MK.ICC](SN.IFD)$
2	B	GET CHALLENGE Compute session keys for authentication as described below: $ZZ = ENC[SK.IFD](RND.ICC RND1.IFD)$ $HASH1 = h(ZZ c)$ with $c=1$ $HASH2 = h(ZZ c)$ with $c=2$ $HASH3 = h(ZZ c)$ with $c=3$ Built key K_{ENC}^a from HASH1 Built key K_{ENC}^b from HASH2 Built key K_{MAC} from HASH3	→ ←	Generate random RND.ICC Compute session keys for authentication as described below: $ZZ = ENC[SK.IFD](RND.ICC RND1.IFD)$ $HASH1 = h(ZZ c)$ with $c=1$ $HASH2 = h(ZZ c)$ with $c=2$ $HASH3 = h(ZZ c)$ with $c=3$ Built key K_{ENC}^a from HASH1 Built key K_{ENC}^b from HASH2 Built key K_{MAC} from HASH3
3	C ⁷	C1: Compute E and M as follows: $E = ENC[K_{ENC}^a](RND.ICC \text{Application-specific payload})$ $M = MAC[K_{MAC}](E)$ EXTERNAL AUTHENTICATE (<E M>)	→ ←	C1: Verify the cryptogram, decrypt and get Application-specific payload value. Optionally, generation of an ephemeral public key pair (PrK.KA.ICC, PuK.KA.ICC) for key agreement (ref. Table 7)
		or C2: Generate Random RND2.IFD INTERNAL AUTHENTICATE (RND2.IFD)		or C2: Compute E and M as follows: $E = ENC[K_{ENC}^b](RND2.IFD \text{Application-specific payload})$

⁶ Refer to Table 1 for P1 parameter distinction.

⁷ If step C1 is operated, the subsequent steps (from 4 to 7) are not executed. If step C2 is operated, the subsequent steps (from 4 to 7) are executed.

stage	step	IFD	Transmission	ICC
		Verify the cryptogram, decrypt and get Application-specific payload value		$M = \text{MAC}[K_{\text{MAC}}](E)$ Return $\langle E \parallel M \rangle$
4	D	Establish context for IFD certificate(s) verification MSE SET DST (CAR) Send the certificate(s) of the IFD (CV) PSO:VERIFY CERTIFICATE(CV) With PuK.GENUINE.IFD = IFD static public key, matching certificate CV.	\rightarrow \leftarrow	Establish context for certificate verification Verify certificate CV
5	E	Get Domain parameters D for Key Agreement Generate a key pair over D (PrK.KA.IFD, PuK.KA.IFD) READ BINARY or GET DATA	\rightarrow \leftarrow	Domain parameters D for Key Agreement (i.e. DH, ECDH)
6	F	Establish context for key agreement MSE SET KAT (none)	\rightarrow \leftarrow	Establish context for key agreement
	G	Compute EPK.IFD and MPK.IFD as follows: $\text{EPK.IFD} = \text{ENC}[K_{\text{ENC}}^a](\text{PuK.KA.IFD} \parallel h(\text{PuK.GENUINE.IFD}))$ $\text{MPK.IFD} = \text{MAC}[K_{\text{MAC}}](\text{EPK.IFD})$ With PuK.KA.IFD = IFD ephemeral public key for KA PuK.GENUINE.IFD = IFD static public key, matching certificate CV.	\rightarrow \leftarrow	Verify MPK.IFD Decrypt EPK.IFD and check its content Compute EPK.ICC and MPK.ICC as follows: $\text{EPK.ICC} = \text{ENC}[K_{\text{ENC}}^b](h(\text{PuK.KA.IFD} \parallel \text{PuK.GENUINE.IFD} \parallel \text{PuK.KA.ICC}))$ $\text{MPK.ICC} = \text{MAC}[K_{\text{MAC}}](\text{EPK.ICC})$ With PuK.KA.ICC = ICC public key for KA PuK.KA.IFD = IFD public key for

stage	step	IFD	Transmission	ICC
		<p>GA : key agreement (<EPK.IFD> <MPK.IFD>)</p> <p>Verify MPK.ICC Decrypt EPK.ICC and check its content</p> <p>Compute shared secret ZZ</p> <p>ZZ= KA(PrK.KA.IFD,PrK.GENUINE.IFD,PuK.KA.ICC) With KA being a function for DH/ECDH shared secret generation</p> <p>Build SM session keys from ZZ and initialize SSC to '00..00'</p>		<p>KA PuK.GENUINE.IFD = IFD static public key, matching certificate CV.</p> <p>Return <EPK.ICC> <MPK.ICC></p> <p>Compute shared secret ZZ</p> <p>ZZ= KA(PrK.KA.ICC,PuK.GENUINE.IFD,PuK.KA.IFD) With KA being a function for DH/ECDH shared secret generation</p> <p>Build SM session keys from ZZ and initialize SSC to '00..00'</p>
7	H	Establish Secure Channel		

Stage 1 – Set the cryptographic context

In this step, the cryptographic context is selected by the IFD.

Table 1— stage 1: MSE SET AT – Command

Command parameter	Meaning
CLA	Acc. ISO 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'81' set for external authentication '41' set for internal authentication
P2	'A4' CRT for Authentication
Lc Field	'xx'
Data Field	'94'-L ₉₄ -SN.IFD RND1.IFD '80'-L ₈₀ -Algorithm identifier for mERA '83'-L ₈₃ -MK.ICC reference

Table 2 — stage 1: MSE SET AT – Response

Response parameter	Meaning
Data Field	empty
SW1-SW2	Refer to ISO 7816-4

The Algorithm identifier for mERA are acc. CEN EN14890 part-1 Annex A.
 The ICC computes SK.IFD, the key associated to the IFD as follows:

$$SK.IFD = KDF [MK.ICC](SN.IFD)$$

The length of RND1.IFD as well as the key derivation function KDF is specified by the Algorithm identifier set in step 1 (see Table 19 —Cryptographic suites
)

Note: This command data field may be expanded to convey in an application-specific way an application specific descriptor.

Stage 2 – Generation of the authentication keys

In this step, the IFD requires the ICC to generate a challenge.

Table 3 — stage 2: GET CHALLENGE – Command

Command parameter	Meaning
CLA	Acc. ISO 7816-4
INS	'84' GET CHALLENGE
P1	'00'
P2	'00'
Lc Field	'xx'
Data Field	empty

Table 4 — stage 2: GET CHALLENGE – Response

Response parameter	Meaning
Data Field	RND.ICC
SW1-SW2	Refer to ISO 7816-4

Once the ICC has returned a challenge, both the ICC and the IFD can compute the session keys.

The shared secret ZZ is computed as follows:

$$ZZ = \text{ENC}[\text{SK.IFD}](\text{RND.ICC} \parallel \text{RND1.IFD})$$

And the session keys are built as follows

$$\begin{aligned} \text{HASH1} &= h(\text{ZZ} \parallel c) \text{ with } c=1 \\ \text{HASH2} &= h(\text{ZZ} \parallel c) \text{ with } c=2 \\ \text{HASH3} &= h(\text{ZZ} \parallel c) \text{ with } c=3 \end{aligned}$$

$$\begin{aligned} &\text{Built key } K_{\text{ENC}}^a \text{ from HASH1} \\ &\text{Built key } K_{\text{ENC}}^b \text{ from HASH2} \\ &\text{Built key } K_{\text{MAC}} \text{ from HASH3} \end{aligned}$$

The construction of keys from HASH1, HASH2 and HASH3 is described in EN 14890 section 8.10.

The length of RND.ICC, the hash function h(), ENC and MAC functions are specified by the Algorithm identifier set in step 1 (see Table 19 — Cryptographic suites)

Stage 3 – External authentication of the IFD (C1)

In this step, the IFD is authenticated and can convey data to the ICC in a way ensuring confidentiality, integrity and authenticity.

Table 5 - stage 3, step C1: EXTERNAL AUTHENTICATE – Command

Command parameter	Meaning
CLA	Acc. ISO 7816-4
INS	'82' EXTERNAL AUTHENTICATE
P1	'00' no specific information
P2	'00' no key reference
Lc Field	'xx'
Data Field	ENC[K ^a _{ENC}](RND.ICC Application-specific payload) MAC[K _{MAC}](ENC[K ^a _{ENC}](RND.ICC Application-specific payload))

Table 6 - stage 3, step C1: EXTERNAL AUTHENTICATE – Response

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

The IFD sends to the ICC the application-specific payload.

Upon reception, the ICC:

- Checks the MAC;
- Decrypts the data;
- Checks RND.ICC;
- Retrieves the application-specific payload;

The ENC and MAC functions are specified by the Algorithm identifier set in step 1(see Table 19 —Cryptographic suites)

Optionally, an ephemeral public key for key agreement (PrK.KA.ICC, PuK.KA.ICC) may be generated on-card upon execution of a GENERAL AUTHENTICATE command (Table 7)

Table 7 —stage 6.G: GA : key generation – Command

Command parameter	Meaning
CLA	Acc. ISO 7816-4
INS	'86' GENERAL AUTHENTICATE
P1	'00'
P2	'00'
Lc Field	'xx'
Data Field	'7C'-L _{7C} '06' L ₀₆ – {OID_mERA} '85' L ₈₅ - { ref to an ephemeral public key for key agreement}

Table 8—stage 6: GA : key generation - Response

Response parameter	Meaning
Data Field	'7C'-L _{7C} '06' L ₀₆ – {OID_mERA} '85' L ₈₅ - '00'
Status Word	Refer to ISO 7816-4

Stage 3 – Internal authentication of the ICC (C2)

In this step, the ICC authenticates itself in an anonymous way and can convey data to the IFD in a way ensuring confidentiality, integrity and authenticity.

Table 9 - stage 3, step C2: INTERNAL AUTHENTICATE – Command

Command parameter	Meaning
CLA	Acc. ISO 7816-4
INS	'88' INTERNAL AUTHENTICATE
P1	'00' no specific information
P2	'00' no key reference
Lc Field	'xx'
Data Field	RND2.IFD
Le Field	'00'

Table 10 - stage 3, step C2: INTERNAL AUTHENTICATE – Response

Response Parameter	Meaning
Data field	ENC[K ^b _{ENC}](RND2.IFD Application-specific payload) MAC[K _{MAC}](ENC[K ^b _{ENC}](RND2.IFD Application-specific payload))

SW1-SW2

Refer to ISO/IEC 7816-4

The IFD generates a random RND2.IFD and sends it to the ICC.

Upon reception, the ICC returns the application-specific payload.

The data are encrypted and protected in integrity as described in the table.

Upon reception, the IFD

- Checks the MAC;
- Decrypts the data;
- Checks RND2.IFD;
- Retrieves the application-specific payload;

In particular this step may be used to return PuK.KA.ICC to the IFD in a manner ensuring confidentiality, integrity and authenticity.

The ICC has a key pair (PuK.KA.ICC, PrK.KA.ICC) where

- PuK.KA.ICC is the public portion of the ICC for key agreement;
- PrK.KA.ICC is the private portion of the ICC for key agreement;

In case of DH, $\text{PuK.KA.ICC} = g^{\text{PrK.KA.ICC}} [p]$

With

- p the large prime modulus;
- q public parameter
- g generator

In case of ECDH, $\text{PuK.KA.ICC} = \text{PrK.KA.ICC} * G$

With

- * being the group multiplication;
- p prime modulus;
- a, b coefficients of the curve
- G being the generator point of order n;
- n order of the base point

The length of RND2.IFD, as well as the ENC and MAC functions are specified by the Algorithm identifier set in step 1 (see Table 19 —Cryptographic suites
)

Stage 4 – Certificate verification

In this stage, the IFD proves being a genuine one. The IFD uses an asymmetric key pair (PrK.GENUINE.IFD, PuK.GENUINE.IFD), whose public key PuK.GENUINE.IFD was certified by a CA known to the ICC. A card verifiable certificate(s) (CVC) CV, was given to

the IFD by the certification authority. That certificate(s) contain(s) PuK.GENUINE.IFD signed by the certification authority and ensures its integrity and authenticity.

First the IFD selects the public key of the certification authority the ICC shall use:

Table 11 — stage 4: MSE SET DST – Command

Command parameter	Meaning
CLA	Acc. ISO 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT
P1	'81' set for verification
P2	'B6' CRT for digital signature
Lc Field	'xx'
Data Field	'83'-L ₈₃ -Name of the Certification authority (CAR)

Table 12 — stage 4: MSE SET DST – Response

Response parameter	Meaning
Data Field	empty
SW1-SW2	Refer to ISO 7816-4

Then the IFD sends CV to the ICC so that it can verify it:

Table 13 —stage 4: Certificate verification – Command

Command parameter	Meaning
CLA	Acc. ISO 7816-4
INS	'2A' PERFORM SECURITY OPERATION
P1	'00' response data field absent
P2	'BE' VERIFY CERTIFICATE
Lc Field	'xx'
Data Field	CV Self descriptive certificate

Table 14 —stage 4: Certificate verification — Response

Response parameter	Meaning
Data Field	empty
SW1-SW2	Refer to ISO 7816-4

The certificate CV is a self descriptive certificate formatted as described in table 18.

If a certificate chain is available, these two commands may be repeated several times.

Once the Certificate is verified, the IFD and ICC can perform a key confirmation and generate a secure channel.

Stage 5 – Retrieval of Public parameters for Key agreement

This stage (and the subsequent) is only performed if step C2 was done.

This may be done by command GET DATA or READ BINARY.

In case DH is used, the domain parameters D are made of (ref. EN 14890-1 clause 11.4 Table-156):

- p the large Prime modulus;
- g the generator;
- q the public parameter;

In case ECDH is used, the domain parameters D are made of (ref. EN 14890-1 clause 11.5 Table-157):

- p the prime modulus;
- G the generator point of order n;
- n the order of the generator;
- The curve, defined by a and b (the coefficient of the curves);

Once the IFD retrieved the Domain parameters used by the ICC, it can generate a key pair over D (PuK.KA.IFD, PrK.KA.IFD) where

- PuK.KA.IFD is the public portion of the IFD for key agreement
- PrK.KA.IFD is the private portion of the IFD for key agreement

In case DH is used, $\text{PuK.KA.IFD} = g^{\text{PrK.KA.IFD}} [p]$

In case of ECDH is used, $\text{PuK.KA.IFD} = \text{PrK.KA.IFD} * G$, with * being the group multiplication.

It shall be noted that the IFD might have already generated the key pairs, and therefore could re use an existing key pair.

Stage 6 –Key agreement

In this stage, the IFD sets the context for key agreement.

Table 15 — stage 6.F: MSE SET KAT – Command

Command parameter	Meaning
CLA	Acc. ISO 7816-4
INS	'22' MANAGE SECURITY ENVIRONMENT

P1	'C1' set for key agreement
P2	'A6' CRT for Key Agreement
Lc Field	'02'
Data Field	'8300' – Key implicitly known

Table 16 — stage 6.F: MSE SET KAT – Response

Response parameter	Meaning
Data Field	empty
SW1-SW2	Refer to ISO 7816-4

It takes place with the command GENERAL AUTHENTICATE used as follows:

Table 17 —stage 6.G: GA : key agreement – Command

Command parameter	Meaning
CLA	Acc. ISO 7816-4
INS	'86' GENERAL AUTHENTICATE
P1	'00'
P2	'00'
Lc Field	'xx'
Data Field	'7C'-L _{7C} '86' L ₈₆ <EPK.IFD> '84' L ₈₄ <MPK.IFD>

Table 18—stage 6: GA : key agreement - Response

Response parameter	Meaning
Data Field	'7C'-L _{7C} '86' L ₈₆ <EPK.ICC> '84' L ₈₄ <MPK.ICC>
Status Word	Refer to ISO 7816-4

The IFD sends EPK.IFD and MPK.IFD to the ICC computed as follows:

$$\text{EPK.IFD} = \text{ENC}[K_{\text{ENC}}^a](\text{PuK.KA.IFD} \parallel h(\text{PuK.GENUINE.IFD}))$$

$$\text{MPK.IFD} = \text{MAC}[K_{\text{MAC}}](\text{EPK.IFD})$$

Where:

- PuK.KA.IFD = the IFD public key for KA;
- PuK.GENUINE.IFD = the IFD static public key, matching certificate CV;

EPK.IFD and MPK.IFD are conveyed within the DOs of the GENERAL AUTHENTICATE command without any further BER-TLV structure.

Upon reception, the ICC

- checks MPK.IFD;
- Decrypt EPK.IFD;
- Checks the content of the plain text data;

When the verification succeeded, the ICC returns EPK.ICC and MPK.ICC computed as follows:

$$\text{EPK.ICC} = \text{ENC}[K_{\text{ENC}}^b](h(\text{PuK.KA.IFD} \parallel \text{PuK.GENUINE.IFD} \parallel \text{PuK.KA.ICC}))$$

$$\text{MPK.ICC} = \text{MAC}[K_{\text{MAC}}](\text{EPK.ICC})$$

Where:

- PuK.KA.ICC = ICC public key for KA;
- PuK.KA.IFD = IFD public key for KA;
- PuK.GENUINE.IFD = IFD static public key, matching certificate CV;

EPK.ICC and MPK.ICC are returned within the GENERAL AUTHENTICATE DOs without any further BER-TLV structure.

In case DH is used, the ICC computes the shared secret as follows:

$$\text{ZZ} = \text{PuK.KA.IFD}^{\text{PrK.KA.ICC}} [p] \parallel \text{PuK.GENUINE.IFD}^{\text{PrK.KA.ICC}} [p]$$

Where

- $\text{PuK.KA.IFD} = g^{\text{PrK.KA.IFD}} [p]$
- $\text{PuK.GENUINE.IFD} = g^{\text{PrK.GENUINE.IFD}} [p]$

In case ECDH is used, the ICC computes the shared secret as follows:

$$\text{ZZ} = \text{COMP}(\text{PrK.KA.ICC} * \text{PuK.KA.IFD}) \parallel \text{COMP}(\text{PrK.KA.ICC} * \text{PuK.GENUINE.IFD})$$

Where

- $\text{PuK.KA.IFD} = \text{PrK.KA.IFD} * G$
- $\text{PuK.GENUINE.IFD} = \text{PrK.GENUINE.IFD} * G$;
- * being the group multiplication;
- COMP being the compression function, returning the X coordinate of the point;

The ICC computes the secure messaging keys from ZZ as described in 14890-1 §8.10 and initializes the SSC to '00'.

The IFD retrieves EPK.ICC and MPK.ICC and

- checks MPK.ICC;
- Decrypt EPK.ICC;
- Checks the content of the plain text data;

In case DH is used, the IFD computes the shared secret as follows:

$$ZZ = \text{PuK.KA.ICC}^{\text{PrK.KA.IFD}} [p] \parallel \text{PuK.KA.ICC}^{\text{PrK.GENUINE.IFD}} [p]$$

Where

- $\text{PuK.KA.ICC} = g^{\text{PrK.KA.ICC}} [p]$;

In case ECDH is used, the IFD computes the shared secret as follows:

$$ZZ = \text{COMP}(\text{PrK.KA.IFD} * \text{PuK.KA.ICC}) \parallel \text{COMP}(\text{PrK.GENUINE.IFD} * \text{PuK.KA.ICC})$$

Where

- * being the multiplication operation over the curve;
- $\text{PuK.KA.ICC} = \text{PrK.KA.ICC} * G$;
- COMP being the compression function, returning the X coordinate of the point;

The IFD computes the secure messaging keys from ZZ as described in 14890-1 §8.10 and initializes the SSC to '00'.

Once the key confirmation is completed, a secure messaging session may be used depending on the requirements of the application.

The Algorithm identifier set in step 1 (see Table 19 —Cryptographic suites) specifies:

- the hash function used for the key derivation function;
- the secure messaging keys type and length;

the ENC and MAC block functions to be used within the secure messaging layer;

Cryptographic suites

The supported cryptographic combinations are described in the table below:

Table 19 —Cryptographic suites

Cipher block E	MAC function	Length of random number	Hashing function	Keys	Key Derivation function	AlgID	OIDs for mERA
TDES CBC with padding M2 (ISO/IEC 9797-1)	Retail Mac (ISO/IEC 9797-1 Algorithm 3 Padding method 2)	8 bytes	SHA-1	TDES EDE2	NIST SP 800-108 (KDF in mode counter) with PRF = CMAC (SP 800-38B), and the associated cipher block function	xx	xx.1
AES CBC with padding M2 (ISO/IEC 9797-1)	AES CMAC with pre padding and Padding method 2 (ISO/IEC 9797-1)	16 bytes	SHA-256	AES – 128 bits	NIST SP 800-108 (KDF in mode counter) with PRF = CMAC (SP 800-38B), and the associated cipher block function	xx	xx.2
AES CBC with padding M2 (ISO/IEC 9797-1)	AES CMAC with pre padding and Padding method 2 (ISO/IEC 9797-1)	16 bytes	SHA-256	AES – 192 bits	NIST SP 800-108 (KDF in mode counter) with PRF = CMAC (SP 800-38B), and the associated cipher block function	xx	xx.3
AES CBC with padding M2 (ISO/IEC 9797-1)	AES CMAC with pre padding and Padding method 2 (ISO/IEC 9797-1)	16 bytes	SHA-256	AES – 256 bits	NIST SP 800-108 (KDF in mode counter) with PRF = CMAC (SP 800-38B), and the associated cipher block function	xx	xx.4

Certificate format

The CVC is formatted as follows:

Table 20 —Format of Card verifiable certificate

Tag	Length	Value			
'7F21'	Var	'7F4E'	Var	Certificate content template	Mandatory
		'5F37'	Var	Signature	Mandatory

The data objects that shall be included in the certificate content template are indicated below:

Table 21 —List of data object within the certificate

Tag	Length	Data element	Presence
'5F29'	'01'	Interchange profile descriptor, e.g. Certificate profile Identifier (CPI)	Mandatory Value of CPI is set to '00'
'42'	'08'	Certification authority reference (CAR)	Mandatory
'5F20'	'0C'	Certificate holder reference (CHR)	Mandatory
'7F4C'	Var	Certificate holder authorization template payload (CHAT)	Conditional (*)
'5F4C'	Var	Certificate Holder Authorization (CHA)	Conditional (*)
'5F24'	'06'	Certificate Expiration Date (CXD)	Mandatory
'7F49'	Var.	Certificate holder public key, e.g. PuK.IFD (constructed DO)	Mandatory
'06'	Var.	Object Identifier (OID) for signature algorithm of certificate holder	Mandatory

Note: further fields (including discretionary DO) may be incorporated in the certificate
 (*) either one of '7F4C' or '5F4C' template shall be present.

Table 22 —Certificate Holder Authorization Template

Tag	Length	Value
'7F4C'	Var	
		'06'-L ₀₆ -{mERA_OID}
		'53'-L ₅₃ -{CHA} discretionary data containing the role identifier