INTERNATIONAL STANDARD

ISO/IEC 19286

First edition
2018-01

Identification cards — Integrated circuit cards — Privacy-enhancing protocols and services

Cartes d'identification — Cartes à circuit intégré — Protocoles et services renforçant la protection des données personnelles

Reference number
ISO/IEC 19286:2018(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by ISO/IEC JTC 1, *Information technology*, SC 17, *Cards and security devices for personal identification.*

Introduction

National and pan-national (e.g. European) privacy regulations require the protection of personal data as well as implicitly linked parameters revealing the identity of the cardholder [see relevant documents in different countries (e.g. EU GDPR, US PIA, Canada PIA or Australian PIA)(see 5.1)].

Privacy-enhancing implementations allow a cardholder to be confident that their sensitive personally identifiable information (PII) is not exposed to an unauthorized environment. Thereby a cardholder may be exposed to an environment that might read sensitive PII from the Integrated Circuit Card (ICC) ahead of any external authentication. Such sensitive PII can be unique parameters of a card (e.g. the Card ID) or personalized parameters of the cardholder and could be linked to the cardholder.

For instance, if the nationality of a cardholder can be identified by the nature of the ICC description parameters (e.g. algorithm ID, if unique for particular country) then a cardholder of a certain nationality could be exposed to observation. An employee identification card, a health insurance card, a passport are typical examples which may require privacy protection.

ICC services ensuring privacy could, for instance, find further applications in the context of user privacy issues in eVoting systems with ICCs and in systems using the environment of Internet of Things as well as access services by means of an ICC.

This document reflects these requirements by harmonized operations and/or services in regard to a corresponding level of privacy. It envisions

— to strengthen common technical measures about privacy-enabling interchange at card edge and to facilitate its adoption,

— to harmonize privacy properties or privacy framework definitions when existing, and

— to address generic technical features related to privacy implementation at card edge (interchange) regardless of the cryptographic mechanisms by considering transactional aspects as asynchronous protocols involving several entities in privacy context.

Identification cards — Integrated circuit cards — Privacy-enhancing protocols and services

1 Scope

This document aims to normalize privacy-enhancing protocols and services by

- using the mechanisms from parts of ISO/IEC 7816 and parts of ISO/IEC 18328 that contribute to security and privacy,
- providing discoverability means of privacy-enabling attributes,
- defining requirements for attribute-based credential handling, and
- identifying data objects and commands for ICCs.

Existing privacy-enhancing protocols available in a generic context are adopted for distributed systems including ICCs. Additionally, existing authentication protocols between an ICC and an external device used for establishing a secure channel are enhanced with privacy protection. Secure communication between an ICC and an on-card device is also considered.

All the protocols and services described in this document contribute to privacy. Annex B describes an example of privacy impact assessments of respective systems.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. 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 7816-8, *Identification cards — Integrated circuit cards — Part 8: Commands and mechanisms for security operations*

ISO/IEC 7816-9, *Identification cards — Integrated circuit cards with contacts — Part 9: Interindustry commands for card and file management*

ISO/IEC 7816-11, *Identification cards — Integrated circuit cards with contacts — Part 11: Personal verification through biometric methods*

ISO/IEC 18328-3, *Identification cards — ICC-managed devices — Part 3: Organization, security and commands for interchange*

3 Terms and definitions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

3.1
access phrase
alpha-numeric string to be captured by interface device to gain access to ICC

EXAMPLE MRZ printed on electronic passports and optically captured by inspection system.

3.2
anonymity
characteristic of information that does not permit a *personally identifiable information principal* (3.22) to be identified directly or indirectly

[SOURCE: ISO/IEC 29100:2011, 2.1]

3.3
attribute
user attribute
quality or characteristic ascribed to someone or something

[SOURCE: NIST SP 800-63-3]

EXAMPLE User name, address, date of birth or assertion about date of birth are user attributes.

Note 1 to entry: Examples of user attributes that can be used to identify natural persons are given in Reference [14].

3.4
attribute integrity
capability of an *attribute* (3.3) to resist to unintended or unauthorized modification

3.5
attribute provider
entity (3.13) that makes *user attributes* (3.3) available

Note 1 to entry: An attribute provider may be an *identity provider* (3.18) or an entity mandated by an identity provider.

3.6
attribute statement
statement or assertion about user attributes comprising predicates over *attributes* (3.3)

EXAMPLE The business case age verification usually does not require information about the user attribute “date of birth” but only the verification if the age is above a specific threshold, i.e. the attribute statement over the “date of birth” saying “is over 21”.

3.7
authentication
provision of assurance in the *identity* (3.17) of an *entity* (3.13)

[SOURCE: ISO/IEC 29115:2013, 3.2]

3.8
authentication protocol
defined sequence of messages between an *entity* (3.13) and a verifier that enables the verifier to perform *authentication* (3.7) of an entity

[SOURCE: ISO/IEC 29115:2013, 3.4]

3.9
credential
set of data presented as evidence of a claimed or asserted *identity* (3.17) and/or entitlements

[SOURCE: ISO/IEC 29115:2013, 3.8]

3.10
domain-specific identifier
attribute (3.3) or *attribute statement* (3.6) over an *identifier* (3.16) of an entity, which carries semantics only in specific domains or contexts

Note 1 to entry: In the literature, such domain-specific identifiers may be also referred to as pseudonyms, domain pseudonyms, context-specific identifiers or sector-identifiers and in pseudonymization (see ISO/IEC 29100 for definition of pseudonymization).

Note 2 to entry: In contrast to *anonymity* (3.2), the *user* (3.32) creates and uses an ambiguous parameter, the pseudonym (e.g. a phantasy name), which is not sufficient for user *identification* (3.15) but is useful to partially recognize and address the user for dedicated communication purpose (e.g. chat room, forum).

3.11
eID-Application
on-card application that manages *user attributes* (3.3) for electronic *identification* (3.15) purposes and controls access to the user attributes

3.12
eID-Server
application running on a local or remote server that enables access to *user attributes* (3.3) managed by an *eID-Application* (3.11)

3.13
entity
something that has separate and distinct existence and that can be identified in a context

[SOURCE: ISO/IEC 29115:2013, 3.10]

3.14
generic attributes
user attributes (3.3) that are not linked to the *terminal domain-specific identifier* (3.29) of the requesting terminal stored in a file and identified by a file identifier

3.15
identification
process of distinguishing an *entity* (3.13) within a given context by the unique association of a set of descriptive parameters

EXAMPLE User attributes are descriptive parameters.

3.16
identifier
data which identifies an *entity* (3.13) in a given context towards another entity

3.17
identity
set of *attributes* (3.3) related to an *entity* (3.13)

[SOURCE: ISO/IEC 29115:2013, 3.13]

3.18
identity provider
trusted actor that issues and/or manages *credentials* (3.9)

Note 1 to entry: In literature, such identity provider is often referred to as identity information provider (see ISO/IEC 24760-1) or credential service provider (see ISO/IEC 29115).

3.19
issuer
entity that is an *identity provider* (3.18) or attribute provider

Note 1 to entry: An issuer may also issue the *token* (3.30).

3.20
mutual authentication
authentication of *identities* (3.17) of *entities* (3.13) which provides both entities with assurance of each other's identity

3.21
password
alpha-numeric string kept secret by the user and used for user verification

3.22
personally identifiable information
PII
any information that (a) can be used to identify the *PII principal* (3.23) to whom such information relates, or (b) is or might be directly or indirectly linked to a PII principal

[SOURCE: ISO/IEC 29100:2011, 2.9]

3.23
PII principal
natural person to whom the *personally identifiable information (PII)* (3.22) relates

[SOURCE: ISO/IEC 29100:2011, 2.11]

3.24
privacy impact assessment
PIA
overall process of identifying, analysing, evaluating, consulting, communicating and planning the treatment of potential privacy impacts with regard to the processing of *personally identifiable information* (3.22), framed within an organization's broader risk management framework

Note 1 to entry: This process is also known as a privacy risk assessment.

[SOURCE: ISO/IEC 29134:2017, 3.7]

3.25
secure channel
communication link between the ICC and the external world that provides confidentiality and/or integrity

3.26
sensitive PII
category of *personally identifiable information (PII)* (3.22), either whose nature is sensitive, such as those that relate to the *PII principal's* (3.23) most intimate sphere, or that might have a significant impact on the PII principal

[SOURCE: ISO/IEC 29100:2011, 2.26]

3.27
service provider
entity (3.13) providing one or more services

3.28
specific attributes
user attributes (3.3) that are stored in data containers each linked to a *terminal domain-specific identifier* (3.29)

3.29
terminal domain-specific identifier
identifier of a terminal which carries semantics only in specific domains or contexts

Note 1 to entry: In contrast to domain-specific identifiers generated by the ICC to provide a pseudonym of the user, a terminal domain-specific identifier links a certain terminal identity to a certain domain (sector).

3.30
token
physical device or digital information, holding *credentials* (3.9), *user attributes* (3.3), *attribute statements* (3.6) and/or other information to be used in authentication procedures

3.31
unlinkability
property that *user's* (3.32) transactions are not linked with other transactions of the same user

3.32
user
natural person who receives and subsequently holds the *token* (3.30) and uses it to assert *user attribute* (3.3) information to relying entities

4 Abbreviated terms and notations

ABC	Attribute-Based Credentials
ADF	Application Dedicated File
APDU	Application Protocol Data Unit
AtP	Attribute Provider
CA-ABC	Chip Authentication based on ABC-based signatures
CA-PSA	Chip Authentication based on Pseudonymous Signature Authentication
CAR	Certificate Authority Reference
CAv2	Chip Authentication version 2 as part of EACv2
CHA	Certificate Holder Authorization
CIA	Cryptographic Information Application
CHAT	Certificate Holder Authorization Template
CHR	Certificate Holder Reference
C-RP	Command Response Pair, i.e. pair of command and response APDU
DF	Dedicated File
EACv2	Extended Access Control version 2 NOTE Extended access control version 1 is defined for EU passports.
EF.ATR	Elementary File for Answer-To-Reset
eID	electronic Identification
eMRTD	electronic Machine Readable Travel Document

ERA	Enhance Role Authentication
GUI	Graphical User Interface
ICC	Integrated Circuit Card
IFD	Interface Device
MRZ	Machine Readable Zone
OID	Object Identifier
PACE	Password Authenticated Connection Establishment
PIA	Privacy Impact Assessment
PII	Personally Identifiable Information
PSC	Pseudonymous Signature for Credentials
SM	Secure Messaging
SP	Service Provider
SW1-SW2	Status Word one and Status Word two

5 General privacy principles

5.1 General

A number of basic principles have evolved [e.g. in ISO/IEC 29100 or Fair Information Practice Principles (FIPPs)[24]] of which this document considers the following general principles as being the most relevant from the ICC technical perspective:

- data minimization;
- user control over user attribute release;
- quality of user attributes.

Those principles have been expressed, among others, in data protection recommendations or legislation for the protection of personal data as well as implicitly linked parameters revealing the identity of the token holder (e.g. the OECD principles[1] for transborder flows of personal data of 2013, the Data Protection Convention and Directive of the Council of Europe[2],[3] (also known as Convention 108) or the PIA Frameworks of ISO[17], the US[4], Canada[5] and Australia[6].

[1] The Recommendation of the OECD Council concerning Guidelines governing the Protection of Privacy and Transborder Flows of Personal Data (July 2013).

[2] Council of Europe, Convention for the protection of individuals with regard to automatic processing of personal data, 1981.

[3] Regulation 2016/679 of the European parliament and of the council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data.

[4] US Department Of Commerce PIA requirement based on Homeland Security Presidential Directive 12 (HSPD-12), Policy for a Common Identification Standard for Federal Employees and Contractors (August 27th, 2004).

[5] Directive on Privacy Impact Assessment, The Treasury Board of Canada Secretariat’s (TBS), April 1 2010.

[6] Office of the Australian Information Commissioner, "Guide to undertaking privacy impact assessments", http://www.oaic.gov.au/privacy/privacy-resources/privacy-guides/guide-to-undertaking-privacy-impact-assessments

This document focuses on security safeguards in the context of ICCs to protect PII against risks such as unauthorized access, use or modification or unintended or inappropriate disclosure.

NOTE ISO/IEC 29100 and FIPPS each list at all 11 and 8 privacy principles, respectively. The application of all these principles in a certain business case requires additional technical, organizational and procedural measures (e.g. at issuer, service provider and identity provider side) that are not addressed by this document.

5.2 Data minimization

Data minimization refers to the property of reducing the amount of PII being transmitted in a given transaction to exactly what is required from the point of view of the underlying business process the data is required for. The data minimization principle can be derived from the purpose specification and the proportionality principles.

The excessive release of user attributes as well as the establishment of non-required linkabilities between transactions is a main problem countering the data minimization principle. For instance, conventional signature schemes, such as RSA, DSA or elliptic curve use certificates, which may allow traceability of transactions if not deployed appropriately.

Realizing data minimization in practice does not only require the use of appropriate technology, but it also requires design of business processes to be data minimizing. Current business processes are defined with only the traditional technologies in mind and thus, have substantial shortcomings in terms of data minimization. Thus, implementing the data minimization principle requires the whole identity system to evolve. This document describes the technology that enables system designers to develop data minimizing solutions.

5.3 User control

User control of the release of user attributes refers to a user's authority over which user attribute is released to which entities. This control over disclosure of attribute information is at the core of European data protection legislation and also of various large R&D projects in the security and privacy domain in Europe[7]. User control is also a key principle in the US-based NSTIC program[8]. Different strengths of user control related to the release of data can be achieved depending on the trust in the reader devices and online versus offline interaction.

The way a solution is realized from a technical perspective determines the degree of user control. A first class of use cases is characterized by the ICC being handed over and used in the device of the attribute recipient, while a second class is characterized by the user using their own device (e.g. computer or phone). As the user should be able to select the user attributes to be released in the transaction, a user's own device may be considered more trusted.

Cryptographic technologies defined in this document can be used to enforce user control for the initial release of user attributes to a service provider. Though, in today's complex value chains of online services, data need to be provided by service providers to third party service providers. Interactions between the service providers and third-party providers are out of scope of this document.

5.4 Data quality

Data quality relates to user attributes being accurate and kept up to date and inaccurate or incomplete attributes are rectified or deleted. The correctness aspect of attributes has relevance in the reduction of cost, increasing efficiencies, and avoiding problems for both processing parties and citizens. Hence, data quality is the combination of organizational and technical measures. The approach adopted is to consider technical mechanisms for updating and improving the quality of data retained. Organizational elements are out of scope of this document.

[7] ABC4Trust Consortium, ABC4Trust Web site, available at: https://abc4trust.eu/ PRIME Consortium, PRIME project web site, 2008, www.prime-project.eu PrimeLife Consortium, Primelife project web site, 2010, www.primelife.eu

[8] NSTIC: National Strategy for Trusted Identities in Cyberspace, available at http://www.nist.gov/nstic/.

Any technology that can convey attributes in integrity-protected form can help improve data correctness and thus data quality. Thus, the widespread use of such technology can reduce data management costs of both public and private sector service providers.

6 Privacy architecture

6.1 General

This clause provides an introductory material for the ICC privacy-enhancing protocols and services. Command and data flow between the ICC and "the external world" is defined for each protocol in the following clauses along with its privacy assessment. The "external world" could be one or more of these entities:

a) an IFD;

b) a GUI controlled by either:

   1) the ICC;

   2) the IFD;

   3) the eID-Server;

c) Service providers;

d) eID-Servers;

e) Attribute providers.

The description of multi-party protocols involving an ICC includes information about what participating entity has rights to access particular ICC data and how the user can control this access. Clause 6 lists the participating entities, categorizes the various data and links data with the respective entities. Moreover, several generic privacy requirements are listed in this clause in order to determine to what extent a particular protocol or sequence of protocols contributes to privacy. Figure 1 gives an overview of the document structure focusing on the protocols, mechanisms, data types and participating entities used.

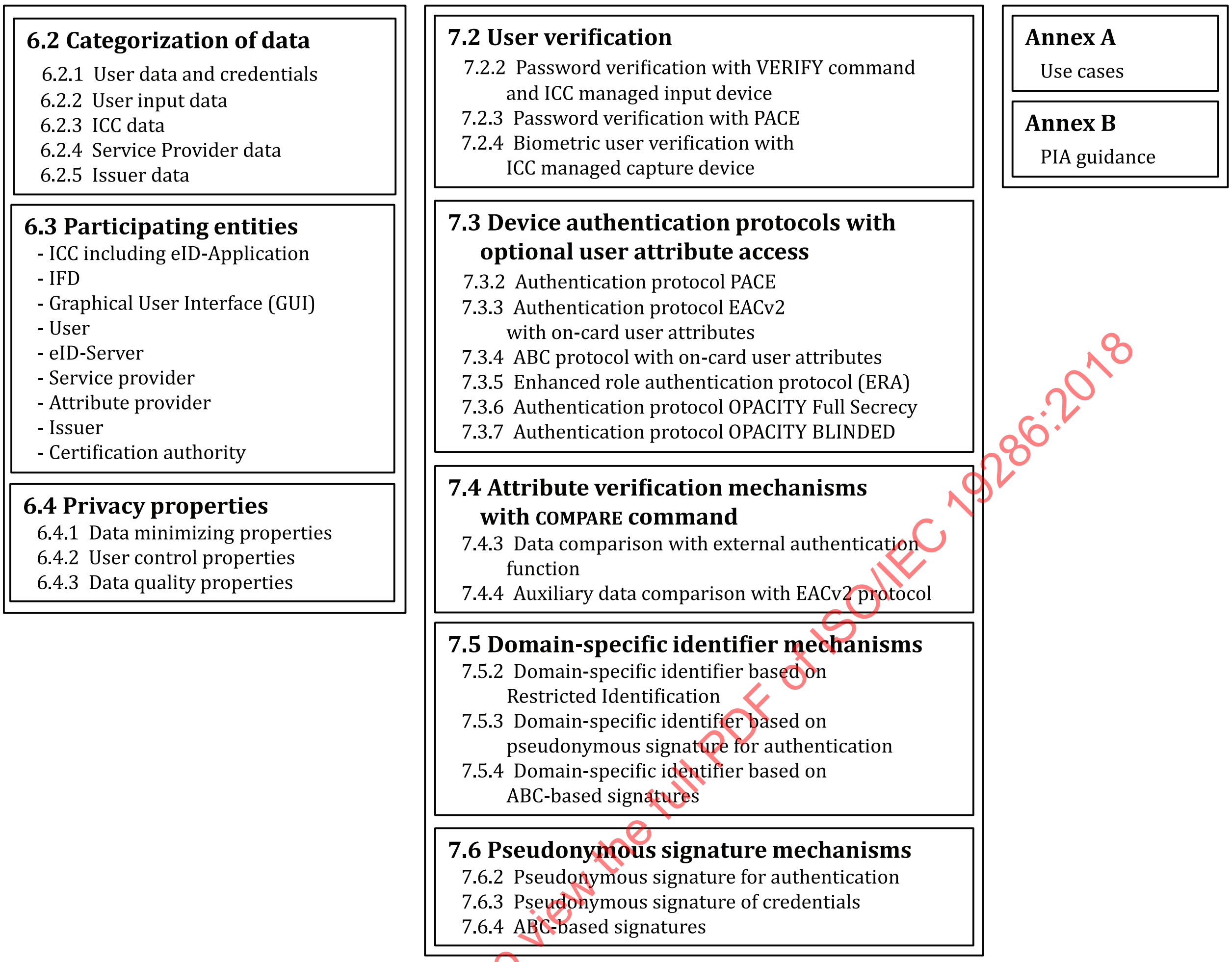
6.2 Categorization of data
6.2.1 User data and credentials
6.2.2 User input data
6.2.3 ICC data
6.2.4 Service Provider data
6.2.5 Issuer data
6.3 Participating entities
- ICC including eID-Application
- IFD
- Graphical User Interface (GUI)
- User
- eID-Server
- Service provider
- Attribute provider
- Issuer
- Certification authority
6.4 Privacy properties
6.4.1 Data minimizing properties
6.4.2 User control properties
6.4.3 Data quality properties
7.2 User verification
7.2.2 Password verification with VERIFY command and ICC managed input device
7.2.3 Password verification with PACE
7.2.4 Biometric user verification with ICC managed capture device
7.3 Device authentication protocols with optional user attribute access
7.3.2 Authentication protocol PACE
7.3.3 Authentication protocol EACv2 with on-card user attributes
7.3.4 ABC protocol with on-card user attributes
7.3.5 Enhanced role authentication protocol (ERA)
7.3.6 Authentication protocol OPACITY Full Secrecy
7.3.7 Authentication protocol OPACITY BLINDED
7.4 Attribute verification mechanisms with COMPARE command
7.4.3 Data comparison with external authentication function
7.4.4 Auxiliary data comparison with EACv2 protocol
7.5 Domain-specific identifier mechanisms
7.5.2 Domain-specific identifier based on Restricted Identification
7.5.3 Domain-specific identifier based on pseudonymous signature for authentication
7.5.4 Domain-specific identifier based on ABC-based signatures
7.6 Pseudonymous signature mechanisms
7.6.2 Pseudonymous signature for authentication
7.6.3 Pseudonymous signature of credentials
7.6.4 ABC-based signatures
Annex A
Use cases
Annex B
PIA guidance

Figure 1 — Overview on the structure of this document without introductory and general clauses

6.2 Categorization of data

6.2.1 User data and credentials

User data (e.g. user attributes or attribute statements or credentials) are PII and refer to any item of personal information identifying the user and delivered under control of an issuing authority. Such data may be stored within an ICC and protected by access rules depending on the issuer's policy. These data may be requested either by an authority eligible to identify the user or by a system granting access to some electronic service (e.g. service provider, identity provider or certification authority).

User attributes

— may be in all or parts of information disclosed to a system upon user consent and upon access condition defined by issuer's policy,

— may be stored in any structure defined in ISO/IEC 7816-4 or in any proprietary structure depending on the application,

— may be replicated on-card of an ICC and stored in a database from where they may be updated onto the ICC,

— may encompass as well the biometric reference data of the user,

— may be permanent or updated over time,

- may be valid for a single access,
- may be valid for a specific time period (e.g. the time period is indicated by the authority that has issued the identity attribute), and
- may be valid until revoked.

Attribute statements can be divided into three main sub-types.

- Domain-specific identifiers (pseudonyms)[9]**:** This sub-type is unique to a domain. A domain-specific identifier may be generated by the user or resulting from a computing process that can take place either within an ICC under user control or on server side.
- Dynamic attribute statements: This may be a certified statement signed by an authority, a statement delivered by a trusted third party under secure conditions or a statement computed by the ICC. These are generated upon request for a service and may be bound to a given service provider.
- Static attribute statements: This represents statements pre-computed by an authority, i.e. not generated in the running course of a transaction. They are not associated to a given service provider and so, they have a generic aspect.

6.2.2 User input data

User input data represent any data deliberately revealed by the user when prompted by a system (e.g. through a human computer interface). This category includes PIN, password or biometric data (e.g. fingerprint data captured on an external or an on-card biometric capture device).

The presentation of user input data (e.g. a password) together with authorization to disclose a list of attributes serves to obtain user consent.

NOTE The very nature of these data can jeopardize the entire privacy measures if not handled with caution by the user (e.g. choosing weak passwords), but they lie outside the boundaries of controlled digital security and can only be mitigated by guidelines and recommendations and by improving users' awareness.

6.2.3 ICC data

This is a category of data stored and retrieved from the ICC and is subject to traceability (e.g. ICC serial number or cryptographic unique identifier). This data is associated with the ICC and may result from its personalization and be set before ICC issuance. This data relates to the identification of a physical support and may be specific to the manufacturer or to the issuer and hence, may allow for indirect user identification. This data may be protected by security attributes specified in ISO/IEC 7816-4 to prevent their disclosure to unauthorized entities.

6.2.4 Service provider data (SP data)

Service provider data is information about the service provider [e.g. the Unique Resource Locator (URL), company name or financial figures]. Parts of this information may be presented to the user by the service provider. The communication protocol used to request a service by the user from the service provider (e.g. the user's computer system or web browser) is out of scope of this document.

6.2.5 Issuer data

Issuer data is any information that identifies the issuer (e.g. electronic signatures, cv-certificates according to ISO/IEC 7816-8 or X.509 certificates according to ITU-T[19] or organization name). This data is typically associated with the ICC and may result from its personalization and be set before

[9] In the literature, such domain-specific identifiers may be also referred to as pseudonyms, domain pseudonyms, context-specific identifiers or sector-identifiers. Moreover, there are also other types of pseudonyms in literature, e.g. transaction pseudonyms.

ICC issuance. Issuer data may allow for indirect user identification and may be protected by security attributes specified in ISO/IEC 7816-4 to prevent their disclosure to unauthorized entities.

6.3 Participating entities

The following participating entities are considered in this document.

a) ICC including eID-Application**:** An ICC may support one or more eID-Applications that manage user attributes and control the access to the user attributes. The user attributes may be written to the ICC, including eID-Application, during personalization or after issuing the ICC to the user. Typically, the eID-Application releases user attributes to a service provider after user authentication.

b) IFD: The IFD generally includes all locally operated components communicating with the ICC. The IFD may:

   1) communicate with the ICC (e.g. according to ISO/IEC 7816-3 or ISO/IEC 14443);

   2) provide means for password entry, biometric user authentication or for displaying information to the user (e.g. a GUI);

   3) be embedded in a hardware device or run as software on a local device (e.g. computer or mobile device) also often referred to as middleware;

   4) manage the communication to one or more remote eID-Servers.

c) Graphical User Interface (GUI): A GUI may display information to the user and may request interaction with the user. A GUI may be controlled by the IFD, by the ICC itself according to ISO/IEC 18328 or by a remote eID-Server.

d) User (see 3.32).

e) eID-Server (see 3.12).

f) Service provider (see 3.27).

g) Attribute provider (see 3.5).

h) Issuer (see 3.19).

i) Certification authority: A certification authority is the entity that certifies a digital identity, typically by issuing digital certificates binding a public key to an identity. An issuer may run a certification authority.

6.4 Privacy properties

6.4.1 Data minimizing properties

6.4.1.1 Partial attribute release

Partial release of user attributes and attribute statements contributes to realizing data minimization. This requires the use of technology, which does not inherently lead to the release of all or large parts of the PII in each transaction.

NOTE Data minimization is closely linked to the principle of “collection limitation” but goes further than that. Whereas “collection limitation” refers to limited data being collected in relation to the specified purpose, “data minimization” strictly minimizes the processing of PII (see ISO/IEC 29100:2011, 5.5).

6.4.1.2 Unlinkability

Unlinkability of transactions at the cryptographic or protocol level may contribute to data minimization. Any identifiers should be made available consciously to establish linkabilities with other transactions as required. Establishing linkability of a transaction with other transactions by default through an improperly designed cryptographic protocol layer counters the ideas of data minimization.

NOTE Today's schemes based on conventional signature schemes lead to linkability. This poses an increasing problem through the use of an eID in a large number of a user's interactions on the Internet.

6.4.1.3 Domain-specific identifier (Pseudonym)

Domain-specific identifiers, or pseudonyms, are another concept towards data minimization. They are a form of identifiers which avoid the use of the same unique identifier for a user in all its interactions. Particularly, when an ICC is used for both governmental applications and private sector applications, some countries mandate different identifiers to be used for public and private sectors. This is to prevent the exposure of substantial personal information from one entity's data to other entities.

6.4.1.4 Pseudonymous signatures

Computing a digital signature based on a domain-specific identifier (pseudonym) and associated attribute statements is a feature relevant for privacy-enhanced interactions where non-repudiation is required. Those signatures can either be an inherent part of the authentication protocol or exposed as a feature for legally-acknowledged digital signatures.

6.4.2 User control properties

6.4.2.1 User-centric system

In a user-centric system, the users have control over the use of their attributes, whether the user attributes are stored within the ICC or managed by an identity provider.

6.4.2.2 Offline/online operation

An online or an offline setting determines whether there is a remote entity (e.g. a service provider) to the transaction or whether the transaction is carried out autonomously without referral to a remote entity. In the offline setting, the user is present at the point where the transaction is executed. In this setting, the ICC may be physically inspected in addition to the use of the digital user attributes. The offline setting typically does not consider the service provider to communicate over a network.

In the online setting, the user performs the transaction remotely over a communication network. Stronger data minimizing and accountability properties are desired in online transactions.

6.4.2.3 Sharing protection

Sharing refers to the voluntary act of the user of giving another person access to the ICC. Thereby, privileges associated with the ICC may be shared with parties who may then be able to illegitimately claim those privileges without holding them. To avoid this, sharing protection can be seen as a basic security requirement. It is crucial that sharing be protected against in order to avoid people claiming privileges they do not hold.

6.4.2.4 User accountability

Accountability of the user is the property that a relying entity, i.e. the service provider, can hold a user accountable for their actions, even if the user is not identified or known under a civil identity at the time of the transaction. Ability to hold user accountable can allow the relying entity to take appropriate actions (e.g. legal investigation or enforcement of its rights). It is important to ensure that such capacity of a system is clearly explained to the user at enrolment and that suitable controls exist that ensure the revelation of an otherwise hidden identity can only occur under specified circumstances.

Those organizational measures are out of scope of this document. A variant of this is that a third-party obtains the identifying statement under the condition and can perform actions. Different trust and execution models may apply for realizing different variants of the user accountability depending on the use case or deployment.

6.4.3 Data quality properties

6.4.3.1 Attribute authenticity

This refers to the authenticity and integrity of attributes being protected and released to the relying entities being consistent with the attributes the issuer has issued. Thus, attribute integrity and authenticity are the basic security properties assuring security of user attribute information not being tampered with. Achieving authenticity and integrity of the contained attributes is a foundational function of any government-issued security document.

6.4.3.2 Civil identity authentication

This refers to the authentication of user attributes or attribute statements corresponding to the civil identity of the user. A civil identity is considered a set of user attributes issued and maintained by a governmental institution. Thus, this is a special case of authenticating attributes with the intention of identifying the user.

NOTE The set of attributes used for identification is often a subset or extended subset of the civil identity of the user to be identified. Many business or government processes require identification, though many of those processes could suffice with less information being revealed, particularly without identifying the user.

6.4.3.3 Verifier accountability

Accountability of the verifier, i.e. the service provider, refers to the responsibility of the verifier of proving to a third-party that a proper verification of certain identity properties of users has been performed. For privacy reasons, this form of accountability should be possible without the users necessarily being identified.

EXAMPLE A merchant is able to prove to third parties intending to check compliance of the merchant, that the minimum age requirement of its customers as stipulated in legal regulations has been verified, without the customers necessarily being identified.

6.4.3.4 User binding

The ICC and, particularly, the credentials should be bound to the user, i.e. the legitimate holder to whom it is issued. This is crucial for the basic function of any government-issued document of associating attributes with people to whom they should apply.

6.4.3.5 Cloning protection

Cloning refers to the illegitimate reproduction of the ICC and user attributes. This may comprise cloning of the physical card body, the digital data stored on the token, or a combination thereof. Cloning may illegitimately give parties using cloned tokens privileges they would not hold otherwise.

6.4.3.6 Eavesdropping protection

Protocols executed between the ICC and a local or remote entity may protect against eavesdropping at the communication and logical layer. The basic security property is to ensure that such eavesdropping does not yield any personal information or any other information that may be useful to an attacker (e.g. information that would allow credential or user attribute cloning).

This is a very basic security property with privacy implications in terms of preventing the leakage of personal information to unauthorized parties. This requirement is crucial for the usage of the ICC in both offline and online settings.

6.4.3.7 Attribute update

The update of attributes refers to the change of attribute values or addition of attributes regardless of whether the attributes are stored within an ICC or on a server. The attribute update can be performed in the field or remotely with or without the holder being required to be in-person.

6.4.3.8 Attribute revocation

Revocation of attributes refers to preventing the revoked attributes from being used in future transactions or ensuring that such use would be recognized as illegitimate by verifiers.

EXAMPLE Revocation of attributes stored in an ISO-compliant driving license can be required in case of the loss of certain driving-related permissions.

7 Privacy-enhancing protocols

7.1 General

This clause describes commands and data flow for interchange of privacy-enhancing protocols and services. These protocols are divided into the following five groups:

— user verification (see 7.2);

— device authentication (see 7.3);

— attribute verification (see 7.4);

— domain-specific identifier (see 7.5);

— pseudonymous signature (see 7.6).

An application designer can choose from the described protocols to best meet their individual requirements. Not every protocol can fulfill the needs of an application and not all protocols can be combined arbitrarily. Selecting an appropriate protocol for a given application is out of scope for this document.

EXAMPLE 1 There are application requirements that do not allow the user attributes to be stored within the ICC but at an attribute provider or vice versa or both, i.e. some user attributes are managed by the ICC whereas other user attributes are managed by one or more attribute providers.

This document focuses on the command and data flow at ICC interface but additionally takes into account the information flow of the external world which allows for addressing privacy issues. The participating entities (e.g. local terminal, remote terminal, attribute/service provider, user or user input/output device) as well as actions to be taken by the entities are named. For instance, it is of importance to describe which entity is in possession of secure messaging keys. Furthermore, how a user can control the information flow related to its user attributes is expressed.

While it is true that the protocols and mechanisms introduced in this document provide certain privacy properties, an application designer deploying one or more of the protocols and mechanisms should be aware that privacy protection requires the entire system to be considered.

EXAMPLE 2 An ICC implements a device authentication protocol with the privacy property of unlinkability and hence, cannot be tracked or traced. If this ICC additionally implements an elementary file readable without any access control and holding a unique serial number, the entire ICC does not fulfill the unlinkability property.

The protocol descriptions are structured in clauses as follows.

a) General protocol description

   This clause describes the main purpose of the protocol and application notes in text format. This clause also lists the fulfilled privacy and security properties (see 6.4).

b) Protocol sequence

This clause describes the overall protocol steps taking all participating entities and the information flow into account in the format of a sequence diagram with the following semantics for the information flow.

⟷ APDU Interface according to ISO/IEC 7816-4 (one or more SM C-RPs)

⟷ SM APDU Interface according to ISO/IEC 7816-4 (one or more SM C-RPs)

⇠⇢ Interface out of scope

c) Data types and relying entities

This clause refines the information flow according to the protocol steps by specifying the type of information as given in 6.2, that is transmitted between a pair of participating entities in the format of a table.

d) C-RP description

This clause specifies the command sequence at ICC interface. Detailed information about data encoding and computations to be taken by the ICC is also specified.

NOTE Specifications are provided for all protocol steps that require actions by the ICC in the format of a C-RP sequence, i.e. the protocol steps marked as line with arrows in normal or bold font in the overall protocol sequence.

e) Protocol-dependent descriptions

This optional clause gives further information about data structures and algorithms related to the protocol.

7.2 User verification

7.2.1 Purpose of user verification

Successful user verification provides some level of assurance that the legitimate user is accessing the ICC. If the ICC fails to authenticate the user, a person may not be the legitimate user and/or the user may use an ICC he is not authorized to use. The success of user verification also provides the user's consent to grant access to credentials, user attributes and/or attribute statements managed by the ICC. User verification should be achieved at the beginning of an authentication process.

7.2.2 Password verification with VERIFY command

7.2.2.1 General protocol description

The user is authenticated by verification of the user password. The password, i.e. the verification data, shall be transmitted to the ICC either in the data field of the VERIFY command or by an ICC-managed input device which may require a sequence of ADDITIONAL DEVICE MANAGEMENT commands according to ISO/IEC 18328-3. If the verification data are transmitted in the data field of the VERIFY command, a device authentication protocol shall be performed prior to command execution (see 7.3).

The password verification is a one-to-one comparison performed by the ICC between the verification data and the reference data stored in the ICC.

If the password verification protocol involves an ICC with an ICC-managed input device, i.e. ICC-managed keypad, the protocol provides user binding and eavesdropping protection property, as well as strong unlinkability.

7.2.2.2 Protocol sequence

This document provides two protocol sequences for password verification with VERIFY command considering an ICC-managed input device (e.g. a keypad) as it is described in ISO/IEC 18328-3. Device handling is either implicit or explicit.

a) Implicit device handling — Password verification with ICC-managed input device and implicit device handling includes the following steps (see Figure 2 without Step 2):

1) IFD transmits VERIFY command with odd INS code and empty optional verification data DO expressing that the verification data comes from an ICC-managed input device;

NOTE ISO/IEC 7816-4 describes that in case of empty verification data DO, the verification data come from a sensor (e.g. fingerprint sensor) on the card. This document considers ICC-managed keypad as input device and uses empty optional verification data DO.

2) user enters password, i.e. the verification data, in the ICC-managed input device;

3) if VERIFY command is successfully executed, security status is set.

b) Explicit device handling — Password verification with ICC-managed input device and explicit device handling includes the following steps (see Figure 2):

1) sequence of additional device management commands (e.g. ADM OPEN DEVICE and ADM MANAGE DEVICE CONFIGURATION) to open and configure ICC-managed input device prior to the execution of VERIFY command;

2) continue with VERIFY command as described in implicit device handling.

NOTE ISO/IEC 18328-3 specifies the command ADDITIONAL DEVICE MANAGEMENT (ADM) for a set of functions which are used to perform all activities of an ICC controlling any ICC-managed device independent of existing or future physical interfaces.

It is out of scope of the protocol descriptions how the user is informed by the IFD or a GUI to present its password, i.e. the verification data. Nevertheless, the protocol sequences can be easily extended by further ADDITIONAL DEVICE MANAGEMENT commands to present a display message to an ICC-managed output device.

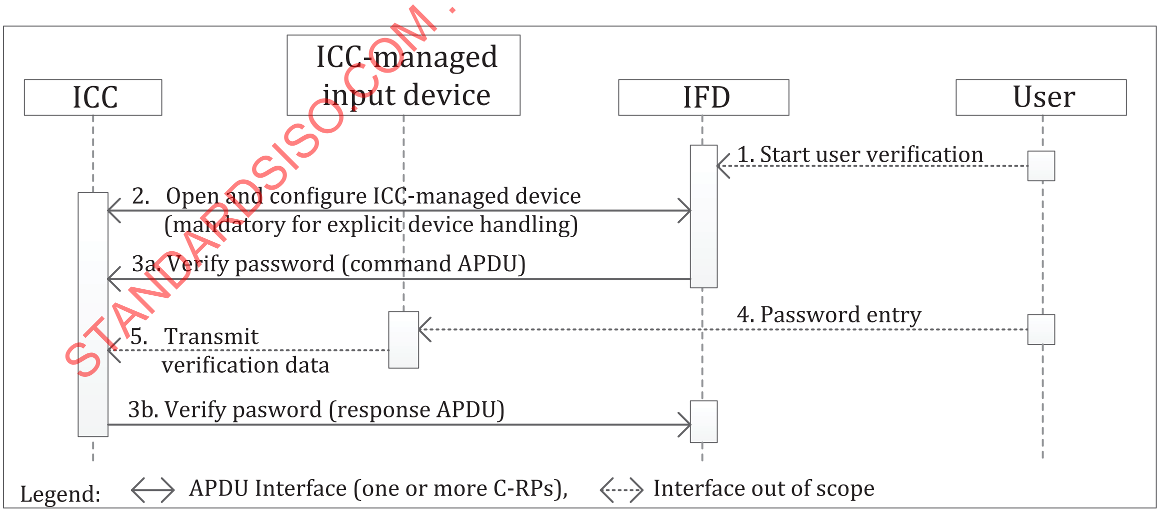
ICC
ICC-managed input device
IFD
User
1. Start user verification
2. Open and configure ICC-managed device (mandatory for explicit device handling)
3a. Verify password (command APDU)
4. Password entry
5. Transmit verification data
3b. Verify pasword (response APDU)
Legend:
APDU Interface (one or more C-RPs),
Interface out of scope

Figure 2 — Password verification with ICC-managed input device

7.2.2.3 Data types and relying entities

Table 1 shows data types and relying entities of password verification with VERIFY command and ICC-managed input device.

Table 1 — Data flow of password verification with ICC-managed input device

Step	Sending entity	Receiving entity	Data type	Description
1	User	IFD	n/a	Start user verification
2	IFD	ICC	ICC data	In case of explicit device handling, the IFD may retrieve general feature management DO on supported services of ICC and sets input device in READY state
3a	IFD	ICC	ICC data	IFD starts password verification with command APDU of VERIFY command and implicitly sets input device in DEVICE OPERATION state
4	User	Input device	User input data	User presents its password to the input device (e.g. by typing the password)
5	Input device	ICC	User input data	Password of the user, i.e. verification data, transmitted to the ICC in a proprietary format
3b	IFD	ICC	ICC data	IFD gets response APDU and verification result from ICC

7.2.2.4 C-RP description

Table 2 shows the C-RP descriptions of password verification with VERIFY command and ADDITIONAL DEVICE MANAGEMENT command.

Table 2 — C-RP sequence of password verification with ICC-managed input device

Step	Command – INS	P1-P2	Command data field, Le field	Response data field, SW1-SW2
2_1	ADDITIONAL DEVICE MANAGEMENT – ‘16’ Function in P1: OPEN DEVICE	‘03 00’	Le=‘01’ – {Device identifier} Device identifier may be obtained from general feature management DO in EF.ATR/INFO; step is mandatory for explicit device handling	{device handle number (DHN)} SWs according to ISO/IEC 18328-3

2_2	ADDITIONAL DEVICE MANAGEMENT – ‘16’
	Function in P1: MANAGE DEVICE CONFIGURATION
		‘0C xx’	‘xx’ – DHN from Step 2_1
			{80 – L – see ISO/IEC 18328-3}
			Configure ICC-managed input device;
			step is optional for explicit device handling
				Data field absent
				SW1-SW2 according to ISO/IEC 18328-3

3a/b	VERIFY – ‘21’	‘00 yy’	‘yy’ – identifier of reference data {‘5F 62’ – ‘00’ – optional verification data DO (empty)} {‘4D’ – L – extended header list referencing verification data DO (optional)} Verify password according to ISO/IEC 7816-4; identifier ‘xx' may be obtained from CIA according to ISO/IEC 7816-15; an empty optional verification data DO expresses that the verification data come from an ICC-managed input device	Data field absent
Password verified, security status set				

7.2.3 Password verification with PACE

7.2.3.1 General protocol description

The PACE protocol[2][25] provides user authentication by deriving cryptographically strong session keys from a password with low entropy entered by the user at IFD side. With the successful establishment of the secure channel between ICC and IFD, the user password has been verified. In contrast, a password verification (e.g. with VERIFY command) requires the transmission of the password to the ICC for one-to-one comparison within the ICC. The PACE protocol does not require the transmission of the password, which makes the protocol especially applicable for ICCs with contactless interface.

Password verification with PACE protocol includes the ICC, the IFD managing an input device and the user. The protocol provides strong eavesdropping protection and unlinkability properties, according to 6.4. Property user binding depends on the security of the input device (e.g. key pad securely managed by the IFD). However, a password can always be shared by the user on purpose.

NOTE A security analysis of the PACE protocol can be found in Reference [20].

If PACE is used in combination with EACv2, the protocol additionally provides confined authorizations of access rights of the local or remote entity. The user may restrict access rights of that entity by means of confined authorizations (see 7.2.3.4.2).

7.2.3.2 Protocol sequence

Figure 3 shows the protocol sequence of password verification with PACE.

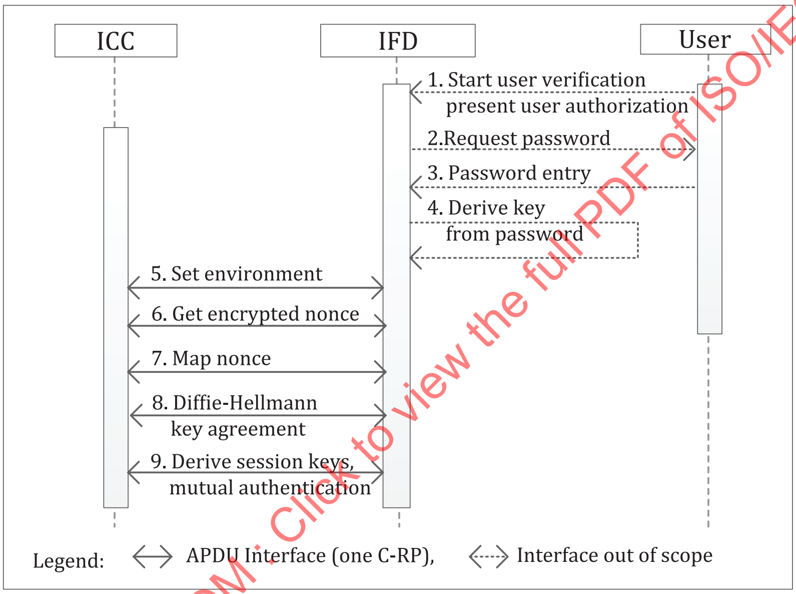

Figure 3 — Password verification with PACE

7.2.3.3 Data types and relying entities

Table 3 shows data types and relying entities of password verification with PACE.

Table 3 — Data flow of password verification with PACE

Step	Sending entity	Receiving entity	Data type	Description
1	User	IFD	User data	Start user verification, conditionally send user CHAT object encoding the restricted access rights by the local or remote terminal (e.g. eID-Server) to be verified in EACv2
2	IFD	User	n/a	Request user password
3	User	IFD	User input data	Present user password by typing password in IFD-managed input device

Table 3 *(continued)*

Step	Sending entity	Receiving entity	Data type	Description
4	IFD	IFD	User input data	Derive PACE-key from user password
5	IFD	ICC	User data	Set protocol parameter and conditionally send user CHAT object
6-9	IFD	ICC	IFD data ICC data	Perform PACE protocol with ephemeral public keys and derive secure messaging keys

7.2.3.4 C-RP description

7.2.3.4.1 C-RP sequence

Table 4 shows the C-RP descriptions of password verification with PACE.

Table 4 — C-RP sequence of password verification with PACE

Step	Command – INS	P1-P2	Command data field, Le field	Response data field, SW1-SW2
5	MANAGE SECURITY ENVIRONMENT – ‘22’ Function P1-P2: SET AT	‘C1 A4’	{‘80’ – L – object identifier for PACE} – {‘83’ – L – password reference} – {‘84’ – L – reference of domain parameter (conditional)} – {‘7F4C’ – L – certificate holder authorization template (conditional) {‘06’ – L – object identifier} – {‘53’ – L – discretionary data object}} Set PACE mode, password reference, domain parameter and conditionally user CHAT object if TAv2 is to be performed after PACE (see 7.3.3 and Table 11).	Data field absent SW1-SW2: according to ISO/IEC 7816-4 and Table 5
6	GENERAL AUTHENTICATE – ‘86’ (command chaining set in CLA byte)	‘00 00’	{‘7C’ – ‘00’ – empty dynamic authentication template} Get encrypted random number	{‘7C’ – L – {‘80’ – L – encrypted random number}}
7	GENERAL AUTHENTICATE – ‘86’ (command chaining set in CLA byte)	‘00 00’	{‘7C’ – L – dynamic authentication template {‘81’ – L – mapping ephemeral IFD public key} Map random number with Generic Mapping	{‘7C’– L – {‘82’ – L – mapping ephemeral ICC public key }}
8	GENERAL AUTHENTICATE – ‘86’ (command chaining set in CLA byte)	‘00 00’	{‘7C’ – L – dynamic authentication template {‘83’ – L – ephemeral IFD public key} Diffie-Hellman key agreement	{‘7C’– L – {‘84’ – L – ephemeral ICC public key }}
9	GENERAL AUTHENTICATE – ‘86’	‘00 00’	{‘7C’ – L – dynamic authentication template {‘85’ – L – IFD authentication token} Perform mutual authentication	{‘7C’– L – {‘86’ – L – ICC authentication token} – {‘87’ – L – CAR$_1$ (cond.)} – {‘88’ – L – CAR$_2$ (cond.)}} See 7.2.3.4.3 SW1-SW2: according to ISO/IEC 7816-4 and 7.2.3.4.3
End of PACE, SM keys derived, Password verified, security status set				

7.2.3.4.2 Step 5 — Set security environment

As the PACE protocol provides password verification, the status words of the MSE SET AT command may give information about password state (see Table 5).

Table 5 — Selection of status words of Step 5

Status words	Meaning
'9000'	Successful operation, protocol parameter set
'63CX'	Warning, Successful operation, X indicates the number of remaining verification tries, if not set to initial value X=0: the password is blocked
'6283'	Warning, Successful operation, password is deactivated

If Terminal Authentication version 2 (see 7.3.3 and Table 11) is to be performed after PACE protocol, confined authorizations are to be sent in a CHAT DO'7F4C', as defined in Table 14. The confined authorizations express the users will to authorize the local or remote entity (e.g. service provider) to get access to user attributes. The CHAT object additionally indicates the terminal type by means of an OID (see Step 9 in 7.2.3.4.3).

7.2.3.4.3 Step 9 — Mutual authentication

The last step of PACE delivers the certificate authority references in DO'87' and DO'88' if TAv2 of EACv2 (see 7.3.3 and Table 11) is to be performed after PACE, i.e. DO'7F4C' (see Table 14) was present in Step 5 of PACE. At most, two trust anchors are to be managed by the ICC, whereas DO'87' holds the most recent trust anchor in terms of certificate effective date.

The PACE protocol provides password verification. The last step of the protocol may give information about the verification and password state (see Table 6).

Table 6 — Selection of status words of Step 9

Status words	Meaning
'9000'	Verification successful
'6300'	Verification failed
'63CX'	Verification failed, X indicates the number of remaining verification tries X=0: the password is blocked

7.2.4 Biometric user verification

7.2.4.1 General protocol description

The user is authenticated by a biometric verification. The biometric data, i.e. the biometric probe, shall be transmitted to the ICC either in the data field of the VERIFY or PERFORM BIOMETRIC OPERATION command according to ISO/IEC 7816-4 and ISO/IEC 7816-11 or by an ICC-managed input device (e.g. an on-card biometric capture device according to ISO/IEC 17839-3) which may require one or a sequence of ADDITIONAL DEVICE MANAGEMENT commands according to ISO/IEC 18328-3. The biometric verification is a comparison of the transferred biometric probe with the biometric reference stored in the ICC.

As the enrolment or verification process requires user interaction, the timing behavior cannot be predicted. A feedback mechanism according to ISO/IEC 17839-3 may be implemented by the ICC and may require further PERFORM BIOMETRIC OPERATION and GET DATA commands.

This protocol recommends an ICC with an ICC-managed input device, i.e. ICC-managed biometric capture device, with implicit and explicit device handling. If an IFD-managed input is applied, a device authentication protocol shall be performed prior to biometric user verification to secure the transmission of the biometric probe from the IFD to the ICC (see 7.3). The protocol provides user binding and eavesdropping protection as well as sharing protection properties according to 6.4.

7.2.4.2 Protocol sequence

Figure 4 shows the protocol sequence of biometric user verification.

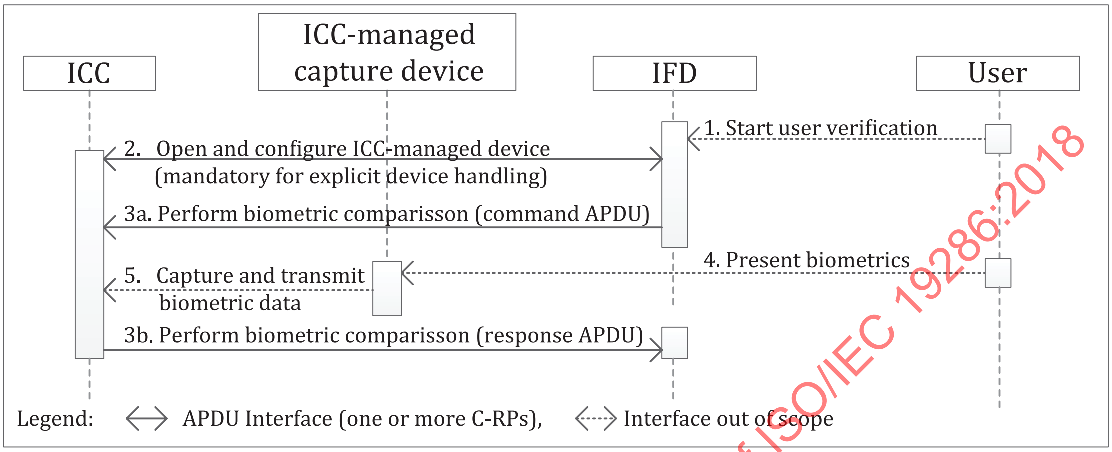
ICC
ICC-managed capture device
IFD
User
1. Start user verification
2. Open and configure ICC-managed device (mandatory for explicit device handling)
3a. Perform biometric comparisson (command APDU)
4. Present biometrics
5. Capture and transmit biometric data
3b. Perform biometric comparisson (response APDU)
Legend: APDU Interface (one or more C-RPs), Interface out of scope

Figure 4 — Biometric user verification with ICC-managed capture device

7.2.4.3 Data types and relying entities

Table 7 shows data types and relying entities of biometric user verification.

Table 7 — Data flow of biometric user verification with ICC-managed capture device

Step	Sending entity	Receiving entity	Data type	Description
1	User	IFD	n/a	Start user verification
2	IFD	ICC	ICC data	In case of explicit device handling, the IFD may retrieve general information of supported services of ICC and sets input device in READY state
3a	IFD	ICC	ICC data	IFD starts biometric verification and sets implicitly input device in DEVICE OPERATION state by sending command APDU of either VERIFY or PERFORM BIOMETRIC OPERATION command
4	User	Input device	User input data	User presents its biometric to the input device (e.g. placing finger on sensor)
5	Input device	ICC	User input data	Biometric data of the user is transmitted to the ICC in a proprietary format
3b	IFD	ICC	ICC data	IFD gets response APDU and verification result from ICC

7.2.4.4 C-RP description

Table 8 shows the C-RP descriptions of biometric user verification.

Table 8 — C-RP sequence of biometric user verification with ICC-managed capture device

Step	Command – INS	P1-P2	Command data field, Le field	Response data field, SW1-SW2
2_1	ADDITIONAL DEVICE MANAGEMENT– ‘16’ Function in P1: OPEN DEVICE	‘03 00’	Le=’02’ – {Device identifier} Device identifier may be obtained from general feature management DO in EF.ATR; step is optional for explicit device handling	{Device handle number (DHN)} SWs according to ISO/IEC 18328-3

2_2	ADDITIONAL DEVICE MANAGEMENT – ‘16’
	Function in P1:
	MANAGE DEVICE CONFIGURATION
		‘0C xx’	‘xx’ – DHN from Step 2_1
			{80 – L – see ISO/IEC 18328-3}
			Configure ICC-managed input device;
			step is optional for explicit device handling
				Data field absent
				SWs according to ISO/IEC 18328-3

3a/b	VERIFY – ‘21’	‘00 yy’	‘yy’ – identifier of biometric reference {‘5F 2E’ – ‘00’ – empty verification data DO} – {‘4D’ – L extended header list referencing verification data DO} Information on biometric verification may be obtained from biometric requirement verification template VIT according to ISO/IEC 7816-4 and ISO/IEC 7816-11; identifier ‘yy’ may be obtained from CIA ccording to ISO/IEC 7816-15; an empty verification data DO expresses that the verification data come from an ICC-managed biometric sensor	Data field absent
Biometric verification done				
Steps 3a and 3b may alternatively performed by Steps $3a_1$, $3a_2$ and 3b				
$3a_1$	MSE SET AT – ‘22’	‘81 A4’	Optional: set usage qualifier, algorithm reference	Data field absent
$3a_2$ or 3b	PERFORM BIOMETRIC OPERATION: Function in P1: COMPARE BIOMETRIC DATA – ‘2A’	‘23 xx’	‘xx’ – identifier of biometric reference { ‘A3’ – ‘00’ empty DO } Compare implicitly selected biometric probe with biometric reference data	Data field absent SWs according to ISO/IEC 7816-11
Biometric comparison done				

7.3 Device authentication protocols with optional user attribute access

7.3.1 Purpose of device authentication protocols

Successful device authentication establishes a relationship of mutual trust between ICC and IFD or remote server. Protocols may also establish a secure communication channel between ICC and IFD or remote server. If device authentication fails, the ICC, IFD or remote server is not trustable and hence, further communication should be denied.

7.3.2 Authentication protocol PACE

7.3.2.1 General protocol description

The PACE protocol[2][25] provides device authentication by using cryptographically strong session keys resulting from a key agreement algorithm and derived from an access phrase with low entropy captured by the device, i.e. the IFD. With the successful establishment of the secure channel between ICC and IFD, the captured access phrase has been implicitly verified. The IFD may optically read the printed access phrase from the card body as it is deployed in electronic passports (eMRTD) or electronic driving licenses according to ISO/IEC 18013. The machine readable zone (MRZ), the scanning area identifier (SAI) or card access number (CAN) serve as PACE access phrase for device authentication. The access phrase may alternatively be manually typed in by the user.

Device authentication is reached by proving that the IFD is in possession of the PACE access phrase that has been captured by using a different information channel than the APDU interface (e.g. optically read). User attributes may be accessible by the IFD after device authentication with PACE.

The protocol provides eavesdropping protection and unlinkability property according to 6.4.

EXAMPLE An inspection system gets access to "less-sensitive data" (see Reference [2]) (e.g. the name, date of birth and facial image) stored in an electronic travel document application, i.e. an eMRTD, after successful authentication with PACE. If the physical passport has been opened and presented to the inspection system by the user or officer for optical capturing of the MRZ and derivation of respective session keys, a basic device authentication is achieved.

NOTE A security analysis of the PACE protocol can be found in Reference [20].

7.3.2.2 Protocol sequence

Figure 5 shows the protocol sequence of device authentication protocol PACE.

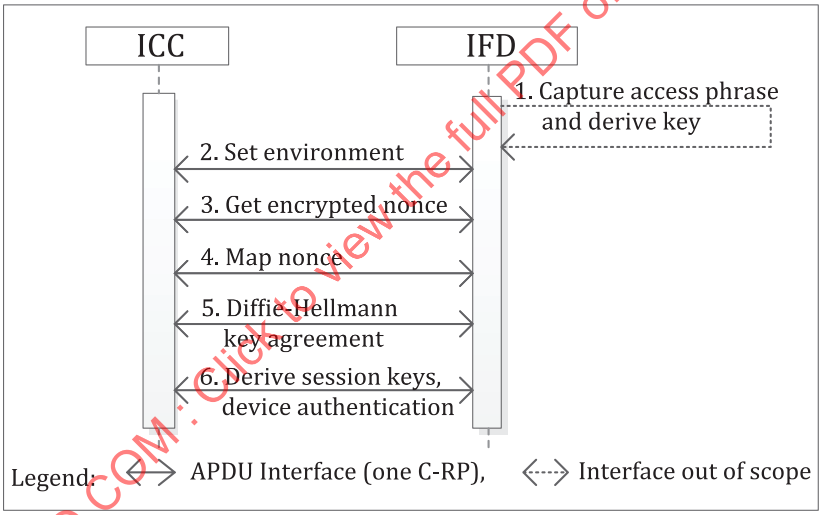
ICC
IFD
1. Capture access phrase and derive key
2. Set environment
3. Get encrypted nonce
4. Map nonce
5. Diffie-Hellmann key agreement
6. Derive session keys, device authentication
Legend:
APDU Interface (one C-RP),
Interface out of scope

Figure 5 — Device authentication with PACE

7.3.2.3 Data types and relying entities

Table 9 shows data types and relying entities of device authentication protocol PACE.

Table 9 — Data flow of PACE device authentication protocol

Step	Sending entity	Receiving entity	Data type	Description
1	IFD	IFD	ICC data	Start device authentication by capturing the access phrase and derive PACE-key from access phrase
2-6	IFD	ICC	IFD data ICC data	Perform PACE protocol with ephemeral public keys and derive session keys

7.3.2.4 C-RP description

See C-RP sequence description in Table 4.

7.3.3 Authentication protocol EACv2 with on-card user attributes

7.3.3.1 General protocol description

The authentication protocol Extended Access Control version 2 (EACv2) requires the user attributes to be managed by an eID-Application resident on the ICC. A mutual authentication between the eID-Application and the eID-Server acting on behalf of a service provider is required by applying a two-party protocol referred to as Chip Authentication (CAv2) and Terminal Authentication (TAv2) while the user may authorize access to its attributes by means of user verification mechanisms (e.g. PIN verification). As the user attributes are managed by the ICC, no remote identity provider is involved which allows for offline access to services of service providers.

The eID-Server needs to be authenticated by the eID-Application, by means of Terminal Authentication as part of the EACv2 protocol (see Reference [25] for Terminal Authentication version 2 and Chip Authentication version 2 as well as ISO/IEC 7816-4:2013, C.2.3).

Terminal Authentication version 2 (TAv2) is divided into two phases. In Phase 1, the public key of the terminal is imported into the eID-Application by validating a chain of cv-certificates starting from the trust anchor stored in the eID-Application. Phase 2 is based on a challenge-response protocol with public-key cryptography. The terminal requests a challenge from the eID-Application and proves possession of the private key by signing the challenge. The eID-Application verifies the signature and challenge by using the corresponding public key of the terminal. Access rights of a particular service provider to the user attributes managed by the eID-Application are coded in the required cv-certificate (see ISO/IEC 7816-8 and 7.3.3.5.1).

NOTE 1 The specific encoding of the access rights in a bit mask as part of the cv-certificates in relation to the logical data structure and user attributes implemented by the eID-Application is out of scope of this document. Nevertheless, the appropriate granularity of the encoding of access rights is crucial to meet privacy requirements. One bit encoding read-access, a second bit encoding write-access for one particular data group of the logical data structure holding one particular user attribute is considered fine-granular. In contrast, one bit encoding read-and-write-access to one single data group holding all user attributes does not allow for dedicated access control by means of cv-certificates.

The eID-Application is authenticated by the corresponding Chip Authentication protocol (see ISO/IEC 7816-4:2013, C.2.3). Chip Authentication version 2 (CAv2) is based on the Diffie-Hellman key agreement protocol and provides authentication of the eID-Application together with establishment of a secure channel. The eID-Application proves possession of a private key to the terminal by the correct derivation of secure messaging keys. The terminal shall additionally validate the corresponding ICC public key by either obtaining it from a trusted source or by reading and verifying it from the eID-Application (e.g. by applying Passive Authentication as specified in Reference [5]).

NOTE 2 If the public key of the eID-Application for CAv2 is individual for each ICC and stored in the ICC as well as accessible without Terminal Authentication, an ICC-individual public key violates the non-linkability privacy property, as it allows for tracing and tracking of the ICC. A non-ICC-individual public key, i.e. the same private/public key pair for CAv2 is shared by a large number of documents, is to be personalized or CA-PSA (see 7.6.2) is to be applied.

The management of the required cv-certificates, static and ephemeral private keys, random numbers and session keys is done by an eID-Server that might be locally operated by the service provider or remotely operated as a service. The communication protocol between the eID-Server and service provider is out of scope of this document.

The EACv2 protocol together with PACE protocol includes the ICC and eID-Application, the IFD managing an input/output GUI, the user, the remote eID-Server and the remote service provider. Moreover, authentication of the issuer requires the deployment of a certification authority in order to perform Passive Authentication. This protocol provides

— partial attribute release through dedicated access rights encoded in the cv-certificate (see NOTE 1 above),

— unlinkability by using different ephemeral values in each protocol run, if a non-ICC-individual private/public key pair or CA-PSA is used in CAv2 (see NOTE 2 above),

— attribute authenticity and eavesdropping protection through end-to-end encryption and authenticity and integrity protection,

— cloning protection through Chip Authentication sub-protocol,

— together with PACE protocol a user-centric system due to required user consent, and

— offline operation possible by locally operated eID-Server and user attributes managed by the ICC.

NOTE 3 A security analysis of the EACv2 protocol can be found in Reference [27].

7.3.3.2 Protocol sequence

Figure 6 shows the protocol sequence of authentication protocol EACv2 with on-card user attributes.

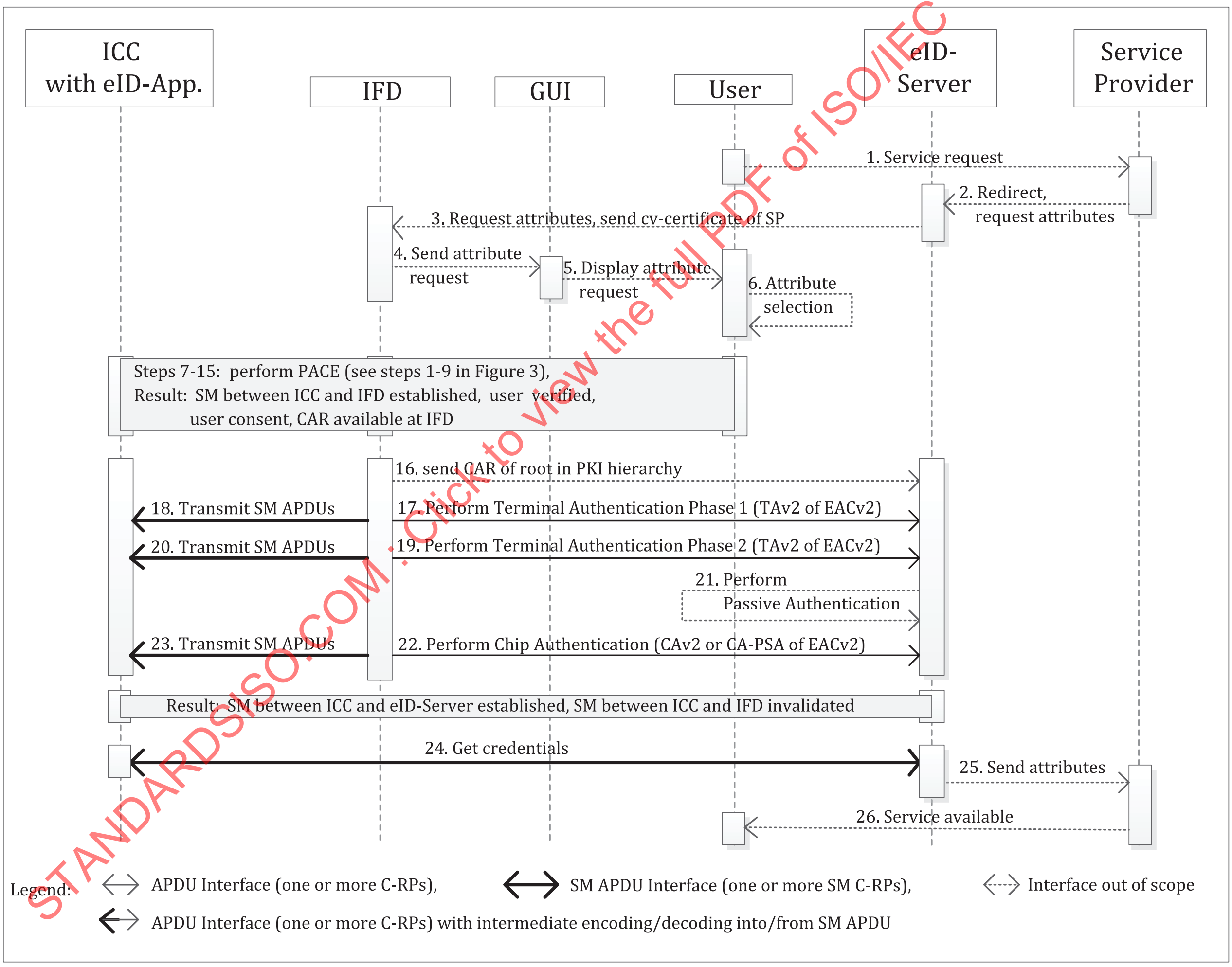
ICC
with eID-App.
IFD
GUI
User
eID-
Server
Service
Provider
1. Service request
2. Redirect,
request attributes
3. Request attributes, send cv-certificate of SP
4. Send attribute
request
5. Display attribute
request
6. Attribute
selection
Steps 7-15: perform PACE (see steps 1-9 in Figure 3),
Result: SM between ICC and IFD established, user verified,
user consent, CAR available at IFD
16. send CAR of root in PKI hierarchy
18. Transmit SM APDUs
17. Perform Terminal Authentication Phase 1 (TAv2 of EACv2)
20. Transmit SM APDUs
19. Perform Terminal Authentication Phase 2 (TAv2 of EACv2)
21. Perform
Passive Authentication
23. Transmit SM APDUs
22. Perform Chip Authentication (CAv2 or CA-PSA of EACv2)
Result: SM between ICC and eID-Server established, SM between ICC and IFD invalidated
24. Get credentials
25. Send attributes
26. Service available
Legend:
APDU Interface (one or more C-RPs),
SM APDU Interface (one or more SM C-RPs),
Interface out of scope
APDU Interface (one or more C-RPs) with intermediate encoding/decoding into/from SM APDU

Figure 6 — EACv2 with on-card user attributes

7.3.3.3 Data types and relying entities

Table 10 shows data types and relying entities of authentication protocol EACv2 with on-card user attributes.

Table 10 — Data flow of EACv2 protocol

Step	Sending entity	Receiving entity	Data type	Description
1	User	SP	n/a	Service request
2	SP	eID-Server	SP data	Forward attribute request to eID-Server
3	eID-Server	IFD	SP data	cv-certificate of SP with requesting user attributes coded in CHA object
4	IFD	GUI	SP data	Attribute request is forwarded to GUI
5	GUI	User	SP data	Present requested attributes of SP to user
6	User	User	SP data	Keep or reduce number of requested attributes in CHA object
7/8/9	User	IFD	User input data	User presents password to the IFD together with user CHA object (see 7.2.3.1)
10-14	IFD	ICC	ICC data	IFD reads protocol-specific data and performs PACE protocol (see 7.2.3.1) with derived key from user password and with user CHA object created in Step 6
15	ICC	IFD	ICC data	SM between ICC and IFD established and transmission of CAR of trust anchor for cv-certificate verification
16	IFD	eID-Server	ICC data	Transmission of CAR of root-CA for cv-certificate verification
17	eID-Server	IFD	SP data	Import SP public key into ICC by verification of chain of cv-certificates
18	IFD	ICC	ICC data SP data	Transform APDUs into SM-APDUs
19	eID-Server	IFD	ICC data SP data	Perform Terminal Authentication by applying challenge-response protocol (challenge received from ICC)
20	IFD	ICC	ICC data SP data	Transform APDUs into SM-APDUs
21	eID-Server	eID-Server	Issuer data ICC data	Perform Passive Authentication, i.e. verify issuer signature of ICC public key including verification of chain of X.509 certificates
22	eID-Server	IFD	ICC data SP data	Perform Chip Authentication by applying Diffie-Hellman key agreement
23	IFD	ICC	ICC data SP data	Transform APDUs into SM-APDUs
24	ICC	eID-Server	ICC data User data	Get credential from ICC, i.e. encrypted and MAC-secured user attributes
25	eID-Server	SP	User data	Send decrypted user attributes to SP by applying any protocol
26	SP	User	n/a	Requested service available to user

7.3.3.4 C-RP description

7.3.3.4.1 Notation

The following notation is used in the specification of the EACv2 protocol.

AUX.data	auxiliary data (see explanation below)
PrK.SP.AUT	private key for authentication of service provider
PuK.SP.AUT	public key for authentication of service provider

PrK.ICC.AUT	private key for authentication of ICC
PuK.ICC.AUT	public key for authentication of ICC
RND1$_{ICC}$	random number generated by ICC
RND2$_{ICC}$	2nd random number generated by ICC
PuK.SP.DH	ephemeral public key of eID-Server (on behalf of service provider)

7.3.3.4.2 C-RP sequence

Table 11 shows the C-RP sequence of authentication protocol EACv2 with on-card user attributes.

Table 11 — C-RP sequence of EACv2 protocol

Step	Command – INS	P1-P2	Command data field, Le field	Response data field, SW1-SW2
PACE has been performed (see 7.2.3.1), SM established, DO'7F4C' processed, CAR$_1$ of root CA is available at IFD				
18a	MSE SET DST – '22' SM activated	'81 B6'	{'83' – L – CAR$_1$} Set key reference of root CA public key	Data field absent
18b	PSO VERIFY CERTIFICATE – '2A' SM activated	'00 BE'	{'7F4E' – L – certificate content template} – {'5F37' – L – signature} Verify self-descriptive cv-certificate and import public key referenced by CHR$_1$	Data field absent
18c	MSE SET DST – '22' SM activated	'81 B6'	{'83' – L – CAR$_2$ equals CHR$_1$ } Set key reference of public key imported in Step 18b and coded in DO'42' CAR in self-descriptive cv-certificate	Data field absent
18d	PSO VERIFY CERTIFICATE – '2A' SM activated	'00 BE'	{'7F4E' – L – certificate content template} – '5F37' – L – signature} Verify self-descriptive cv-certificate and import public key referenced by CHR$_2$	Data field absent
Proceed with cv-certificate import according to PKI hierarchy until IFD public key is available in ICC				
20a	MSE SET AT – '22' SM activated	'81 A4'	{'80' – L – OID of TAv2-protocol} – {'83' – L – CHR$_2$ of public key in Step 18d} – {'91' – L – compressed ephemeral public key for Chip Authentication in Step 23b}– {'67' – L – AUX.data (optional)} Set key reference of SP public key for Terminal Authentication and generate ephemeral public key of SP for Chip Authentication	Data field absent
20b	GET CHALLENGE – '84' SM activated	'00 00'	{command data field absent} Le = '08'	{RND1$_{ICC}$} 8 Bytes random number
20c	EXTERNAL AUTHENTICATE –'82' SM activated	'00 00'	{SIGN(data,PrK.SP.AUT)} Perform challenge-response protocol, for construction of data to be signed see 7.3.3.4.3	Data field absent
End of Terminal Authentication				

Table 11 *(continued)*

Step	Command – INS	P1-P2	Command data field, Le field	Response data field, SW1-SW2
23a	MSE SET AT – '22' SM activated	'41 A4'	{'80' – L – OID of CA-protocol} – {'84' – L – reference of private key of ICC PrK.ICC.AUT} Set protocol parameter and key reference of ICC private key for Chip Authentication	Data field absent
23b	GENERAL AUTHENTICATE –'86' SM activated	'00 00'	{'7C' – L – dynamic authentication data {'80' – L – ephemeral public key of SP, see Step 20a}} Perform DH key agreement and derive SM keys (see 7.3.3.4.6).	{'7C' – L – {'81' – L – RND2$_{ICC}$} – {'82' – L – MAC$_{ICC}$}} See 7.3.3.4.6
End of Chip Authentication, new SM keys derived, effective access rights set				
24a	SELECT –'A4' SM activated	'xxxx'	Select application according to ISO/IEC 7816-4	according to ISO/IEC 7816-4
24b	READ BINARY –'B0' –'B1' SM activated	'xxxx'	Read user attributes according to ISO/IEC 7816-4 and according to effective access rights	according to ISO/IEC 7816-4
Credentials received from ICC				

7.3.3.4.3 Step 20a — Set security environment

The compressed format of a public key embedded in DO'91' shall be either the SHA1 value of the public component for prime fields or the x-coordinate of an elliptic curve public key. Auxiliary data embedded in DO'67' shall be one or more discretionary data templates embedding an object identifier and a discretionary data object. Content and format of the discretionary data object is defined by the OID (see Table 12).

Table 12 — Structure of auxiliary data

'67'	Authentication data template		
	'73'	Discretionary data template	
		'06'	Object identifier defining content and structure of DO
		'53'	Discretionary data object holding information according to OID
	'73'	Further discretionary data templates	
		...	
	...		

7.3.3.4.4 Step 20b — Get challenge

The eID-Server requests a challenge from the ICC with a GET CHALLENGE command to perform the Challenge-Response protocol between ICC and eID-Server.

Step 20b, i.e. GET CHALLENGE command, may be alternatively performed before Step 18a (e.g. in Step 16) to allow for efficient communication between IFD and eID-Server in the overall protocol. The ICC shall keep the challenge valid until Step 20c, except if another GET CHALLENGE command has been performed.

7.3.3.4.5 Step 20c — External Authentication

The data to be signed by the terminal are the concatenated values of the compressed format of the public key used by the ICC in the Diffie-Hellman key agreement of the previous PACE protocol (see Step 8 in Figure 3), the received challenge RND1$_{ICC}$ in Step 20b, the compressed format of the public key of the terminal to be used in the Diffie-Hellman key agreement in Step 23b of Chip Authentication and the authentication data template embedding auxiliary data if transmitted in Step 20a:

data = (comp(PuK.ICC.DH2) || RND1ICC || comp(PuK.IFD.DH) || [DO‘67’]).

7.3.3.4.6 Step 23b — DH key agreement and session key derivation

The eID-Server authenticates the ICC by performing a Diffie-Hellmann key agreement and a new session key derivation. The eID-Server uses an ephemeral public key pair already created in Step 20a. The ICC uses a static key pair.

DH	eID-Server calculates	K_{seed} = PuK.ICC.AUT × PrK.SP.DH mod p
	ICC calculates	K_{seed} = PuK.SP.DH × PrK.ICC.AUT mod p
ECC:	eID -Sever calculates	K_{seed} = Comp(PrK.SP.DH · PuK.ICC.AUT)
	ICC calculates	K_{seed} = Comp(PrK.ICC.AUT · PuK.SP.DH)

r · P is the scalar multiplication between integer r and point P of the elliptic curve

K_{enc} = HASH(K_{seed} || RND2$_{ICC}$ || 1) and K_{mac} = HASH(K_{seed} || RND2$_{ICC}$ || 2) with the bit length of the output of the HASH function shall be equal or greater than the bit length of the key to be derived and determined by the OID for Chip Authentication.

The computation of the authentication token is MAC$_{ICC}$ = (PuK.ICC.AUT, Kmac).

7.3.3.5 Protocol-dependent descriptions

7.3.3.5.1 CV-certificate description

In the protocol EACv2, self-descriptive card-verifiable certificates (see ISO/IEC 7816-8) shall be used for public key import. The data elements given in Table 13 shall be embedded in the certificate body within DO‘7F4E’ in the given order. The certificate body including the template shall be signed. The signature is embedded in DO‘5F37’.

Table 13 — Data objects of certificate body

TAG	Data element
‘5F29’	Certificate profile indicator, default value is ‘00’
‘42’	Issuer identification number, also referred to as Certificate Authority Reference (CAR)
‘7F49’	Public key template; see ISO/IEC 7816-8
‘7F20’	Cardholder name, also referred to as Certificate Holder Reference (CHR)
‘7F4C’	Certificate holder authorization template (CHAT) (see Table 14)
‘5F25’	Certificate effective date
‘5F24’	Certificate expiration date
‘65’	Cardholder-related data, i.e. certificate extensions [see Table 15 (optional)]

Table 14 — Structure of CHA template

‘7F4C’	Certificate holder authorization template (CHAT)	
	‘06’	Object identifier defining CHA
	‘53’	Discretionary data defining Certificate Holder Authorization (CHA) as sequence of byte string, interpretation of bitmask is given by OID

The content of the certificate extension shall be one or more discretionary data templates holding one or more context-specific data objects that are defined by one object identifier, respectively (see Table 15).

Table 15 — Structure of certificate extensions

‘65’	Cardholder related data, i.e. certificate extensions		
	‘73’	Discretionary data template	
		‘06’	Object identifier defining content and structure of DO
		…	one or more context-specific DOs according to OID
	‘73’	Further discretionary data templates according to first DDT	
		…	
	…		

7.3.3.5.2 Computation of effective access rights

When the EACv2 protocol has been completed and a secure channel has been established, see Steps 22/23 in Figure 6. The ICC shall grant access to user attributes or services according to the computed effective access rights. The computation of the effective access rights is the bitwise logical AND computation of all CHA objects along the certificate chain and the user CHA object transmitted and authenticated with the PACE protocol. If a cv-certificate encodes further CHA objects in the certificate extension in addition to the CHA object embedded in the CHAT of the certificate body, a resulting CHA object shall be computed in accordance with the given OID prior to the computation of the effective access rights.

7.3.4 ABC protocol with on-card user attributes

7.3.4.1 General protocol description

The Attribute-Based Credential protocol (ABC protocol) authenticates the eID-Server to the ICC and the ICC to the eID-Server and reveals attributes as authorized by the cv-certificate of the eID-Server from the ICC to the eID-Server. The ABC protocol uses the PACE protocol and Terminal Authentication protocol of EACv2 and leverages the corresponding infrastructure. The CAv2 protocol is replaced by CA-ABC Chip Authentication protocol. The message from the eID-Server for initiating Chip Authentication needs to comprise a random number for ensuring freshness and optionally, an epoch for credential revocation.

Attribute-Based Credentials (ABCs) are a form of privacy-preserving certificates based on specific signature schemes, such as the SRSA-CL signature scheme (see Reference [22]). Conceptually, an ABC is similar to traditional certificates in that it comprises attributes and a signature over the attributes. An ABC allows one, once issued into an ICC, to prove that the ICC contains such ABC without revealing the attributes and signature of the ABC. On the contrary, the ICC proves, using zero-knowledge proof of knowledge protocols, that it holds the ABC without revealing any further information. Particularly, the proof protocols do not reveal the signature of the ABC and allow the ABC to reveal parts of the attribute information comprised in the ABC. The verifier of the proof, that is the eID-Server, is able to obtain cryptographic assurance that the ICC holds the ABC with a valid signature without revealing it.

NOTE A security analysis of the cryptographic signature mechanism related to this protocol has been provided in Reference [22]. The entities and roles of the entities are defined as in the mutual authentication protocol using EACv2 of 7.3.3.3. Each ICC holds its own private key and a sufficiently large number of ICCs share a public key for Chip Authentication based on ABCs. Particularly, there is a non-ICC-individual private key on the ICC, as in CAv2 of EACv2, but those are ephemeral keys, i.e., generated per transaction.

The ICC comprises an ABC with a subset, A_{ABC}, of the ICC-contained attributes. Further attributes, A_{EID}, may be securely and permanently stored in the eID-Application of the ICC. The union of A_{ABC} and A_{EID} is the set of ICC-contained attributes available for being authenticated to an eID-Service and service provider. The sets A_{ABC} and A_{EID} may have a non-empty intersection if required from a deployment perspective. When authenticating attributes in the set A_{ABC} to the eID-Server, this authentication is based on a cryptographic signature, or more precisely, a cryptographic proof of knowledge of such signature. Attributes in the set A_{EID} are not protected by a signature and are securely stored on the

ICC. A comprehensive specification and formal analysis of ABC protocols (also referred to as privacy-enhancing attribute-based credential systems) can be found in Reference [23] as well as in the published results of the ABC4Trust[10] project.

The ABC protocol together with PACE protocol includes the ICC and eID-Application, the IFD managing an input/output GUI, the user, the remote eID-Server and the remote service provider. Moreover, authentication of the issuer requires the deployment of a certification authority in order to perform Passive Authentication. The protocol provides:

- partial attribute release through dedicated access rights coded in the cv-certificate and the release of attribute statements;
- unlinkability by using different values in each protocol run;
- attribute authenticity and eavesdropping protection through end-to-end encryption and authenticity and integrity protection;
- cloning protection through the Chip Authentication protocol;
- together with the PACE protocol, a user-centric system due to required user consent;
- offline/online operation by having the ABC securely contained in the ICC and the identity provider not involved;
- verifier accountability by allowing a third-party verifier (e.g. auditor) besides the original verifier to verify the ABC-based signature;
- user accountability by the ICC encrypting an identifying subset of attributes or attribute predicates accordingly.

7.3.4.2 Protocol sequence

Figure 7 presents a sequence diagram for the protocol flow between the participants, while abstracting the PACE protocol (see 7.2.3) and EACv2 Terminal Authentication protocol (see 7.3.3).

In Steps 21 to 23, the Chip Authentication using CA-ABC protocol (see Table 17) and preceding key authentication is executed. In Step 21, the eID-Server authenticates the public key of the issuer of the ABC used for Chip Authentication. In Step 22, the eID-Server initiates the Chip Authentication protocol CA-ABC by sending an authentication request comprising a random number as well as an epoch. The random number ensures freshness and non-transferability of the proof generated later in the protocol by the ICC in the scope of Chip Authentication. The epoch provides relevant information for the cryptographic credential revocation protocol.

Technically, Chip Authentication comprises the ICC computing a cryptographic zero-knowledge proof that proves to the eID-Server that the ICC holds a valid ABC. The protocol also comprises the eID-Server verifying the cryptographic proof. The protocol can optionally allow the ICC to release the subset of the attributes of the ABC that the service provider may access as specified in its cv-certificate as part of the proof. Those attributes are integrity-protected through the zero-knowledge proof.

[10] Research and development project funded by the European Commission, ABC4Trust Consortium, ABC4Trust Web site: https://abc4trust.eu/.

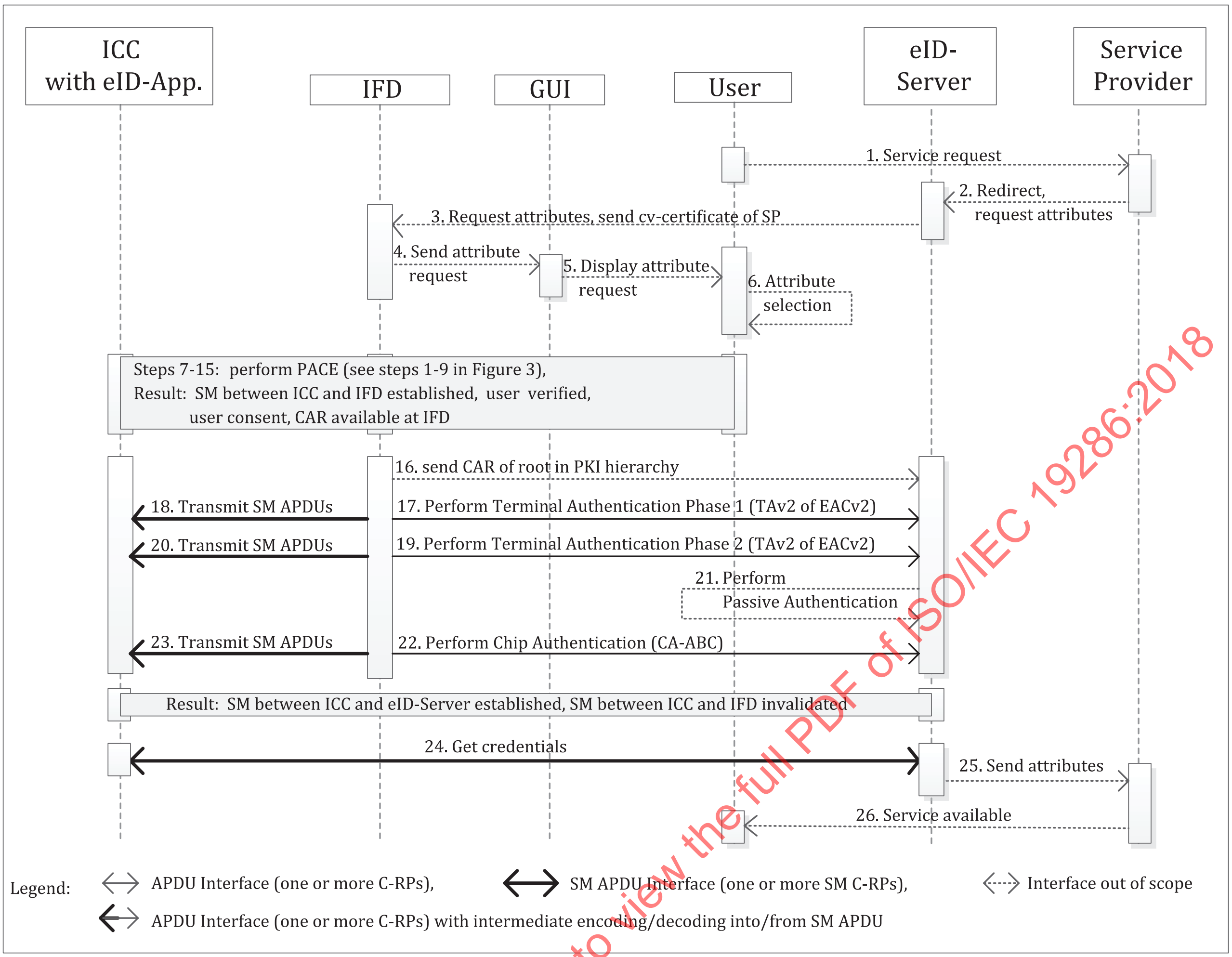
ICC with eID-App.
IFD
GUI
User
eID-Server
Service Provider
1. Service request
2. Redirect, request attributes
3. Request attributes, send cv-certificate of SP
4. Send attribute request
5. Display attribute request
6. Attribute selection
Steps 7-15: perform PACE (see steps 1-9 in Figure 3),
Result: SM between ICC and IFD established, user verified,
user consent, CAR available at IFD
16. send CAR of root in PKI hierarchy
18. Transmit SM APDUs
17. Perform Terminal Authentication Phase 1 (TAv2 of EACv2)
20. Transmit SM APDUs
19. Perform Terminal Authentication Phase 2 (TAv2 of EACv2)
21. Perform Passive Authentication
23. Transmit SM APDUs
22. Perform Chip Authentication (CA-ABC)
Result: SM between ICC and eID-Server established, SM between ICC and IFD invalidated
24. Get credentials
25. Send attributes
26. Service available
Legend:
APDU Interface (one or more C-RPs),
SM APDU Interface (one or more SM C-RPs),
Interface out of scope
APDU Interface (one or more C-RPs) with intermediate encoding/decoding into/from SM APDU

Figure 7 — ABC authentication protocol

7.3.4.3 Data types and relying entities

Table 16 shows data types and relying entities of ABC protocol with on-card user attributes.

Table 16 — Data flow of ABC protocol

Step	Sending entity	Receiving entity	Data type	Description
See Table 10 for Steps 1 to 20.				
21	eID-Server	eID-Server	Issuer data ICC data	Perform Passive Authentication, i.e. verify issuer signature of ABC issuer public key for ABC signature verification including verification of chain of X.509 certificates.
22	eID-Server	IFD	ICC data SP data	Perform CA-ABC by applying Diffie-Hellman key agreement with exchanged ephemeral public keys of ICC and eID-Server, restart secure messaging and authenticate the ICC to the eID-Server using an ABC-based signature computed by the ICC and verified by the eID-Server over the ICC ephemeral public key.
23	IFD	ICC	ICC data SP data	Transform APDUs into SM-APDUs.
See Table 10 for Steps 24 to 26.				

7.3.4.4 C-RP description

7.3.4.4.1 C-RP sequence

Table 17 shows the C-RP sequence of ABC protocol with on-card user attributes.

Table 17 — C-RP sequence of ABC protocol

<table>
<tr><th>Step</th><th>Command – INS</th><th>P1-P2</th><th>Command data field, Le data field</th><th>Response data field, SW1-SW2</th></tr>
<tr><td colspan="5">PACE and Terminal Authentication have been performed (see Steps 18a-d, 20a-c in Table 11)</td></tr>
<tr><td>23a</td><td>MSE SET AT – ‘22’
SM activated</td><td>‘41 A4’</td><td>{‘80’ – L – OID of CA-ABC protocol} –
{‘84’ – L – reference of private key PrK.ICC.AUT}
Set protocol parameter and private key object for Chip Authentication</td><td>Data field absent</td></tr>
<tr><td>23b</td><td>GENERAL AUTHENTICATE –‘86’
SM activated</td><td>‘00 00’</td><td>{‘7C’ – L – dynamic authentication data
{‘80’ – L – ephemeral public key of SP}}
Create ICC ephemeral key for CA-ABC and perform DH key agreement</td><td>{‘7C’ – L –
{‘81’–L–PuK.ICC.AUT)}
Send ICC ephemeral public key for CA-ABC</td></tr>
<tr><td>23c</td><td>MSE SET AT – ‘22’
SM activated</td><td>‘41 A4’</td><td>{‘80’ – L – OID of ABC-based signature protocol}
Set protocol parameter for ABC-based signature computation</td><td>Data field absent</td></tr>
<tr><td>23d</td><td>GENERAL AUTHENTICATE – ‘86’
SM activated</td><td>‘00 00’</td><td>{‘7C’ – L – dynamic authentication data
{‘80’ – L – SP public key for domain-specific identifier (optional)}
{‘90’ – L – random number}
{‘91’ – L – current epoch (optional)}}
Create signature and optionally compute domain-specific identifier(s)</td><td>{‘7C’ – L –
{‘82’–L–PuK.ID.Sector1 or transaction identifier (cond.)}–
{‘83’–L–PuK.ID.Sector2 (cond.)} –
{‘84’–L–(c||s1||s2)}}
Pseudonymous signature c and conditionally the domain-specific identifiers s1 and s2 or conditionally a transaction-specific identifier s1</td></tr>
<tr><td colspan="5">Proceed with Steps 24a-b in Table 11</td></tr>
</table>

7.3.4.4.2 Steps 23a, 23b — Setting protocol parameters and creating ephemeral key pair

In Step 23a, the reference to the private key is sent in a MANAGE SECURITY ENVIRONMENT command to set the private key to be used in Chip Authentication. In contrast to CAv2 described in 7.3.3, the CA-ABC protocol requires the ICC to generate an ephemeral key pair. Hence, the referenced private key does not hold any key data in Step 23a and the ephemeral key data are generated only later in Step 23b. The ephemeral key pair as well as the key agreement stage of the CA-ABC protocol is equivalent to the one of EACv2 using Pseudonymous Signatures for Authentication (CA-PSA) of EAC (see 7.6.2).

7.3.4.4.3 Steps 23c, 23d — Computing ABC-based signature

The ABC-based signature computed in Step 23d is based on the mechanisms specified in References [22] and [23]. The signature generation process may optionally reveal attribute information, thus dynamically determine the membership of a signer group.

The signature generation is preceded by setting the algorithm used for computation of the ABC-based signature in Step 23c. The ABC-based signature computation performed in Step 23d includes generating either one or two domain-specific identifiers if a sector public key of the service provider is provided. In case no such sector public key is provided, a transaction-specific identifier is generated which is valid for the current session and cryptographically unlinkable to all transaction-specific identifiers generated by this ICC in any other session.

A domain-specific identifier or transaction-specific identifier generated in this protocol may be used later in the session for computing a signature over a message or over user attributes. Step 23d also communicates a random number required for security of the signature being computed by the ICC subsequently. The current epoch optionally provided is used for the cryptographic protocol for revocation checking.

7.3.5 Enhanced Role Authentication protocol (ERA)

7.3.5.1 General protocol description

The ERA protocol is a three-party protocol between the ICC, the eID-Server acting on behalf of the service provider and an attribute provider that comprises user verification with PACE (see 7.3.2) and device authentication with EACv2 (see 7.3.3) complemented with "Enhanced Role Authentication" (see References [25] and [26]). In this protocol, a service provider writes user attribute requests in the eID-Application resident on the ICC in case additional user attributes are not available in the eID-Application but that are required to deliver the service. The request is to be responded by an attribute provider. The reading and writing operations by the service provider and attribute provider require authorizations, which are achieved by the Terminal Authentication protocol as part of EACv2.

The eID-Server operated by the service provider and attribute provider are the remote parts of the authentication and attribute terminals. They are authorized to access ICC data and contain the interfaces to the IFD and to the public key infrastructure. The eID-Server provides the IFD with a chain of Terminal Authentication cv-certificates and a digital signature created on the ICCs challenge with the corresponding private key. Enhanced Role Authentication enables the service provider to request additional attributes not yet existing on the ICC to be stored on the ICC by an attribute provider. The following classes of attributes are defined: specific attributes and generic attributes.

The IFD is the local part of the authentication and attribute terminal and interacts with the user, the ICC, and the eID-Server, but is not authorized to access ICC data. In particular, the IFD contains an eID-Client software, an ICC reader, a display and an interface for user input. The Terminal Authentication card-verifiable certificates received from the respective eID-Server are displayed to the user and only if the user accepts, the IFD forwards the received certificates to the ICC.

The ERA protocol, together with EACv2 and PACE protocol, includes the ICC and eID-Application, the IFD managing an input/output GUI, the user, the remote eID-Server and the remote service provider. Moreover, authentication of the issuer requires the deployment of a certification authority in order to perform Passive Authentication. This protocol provides (according to 6.4)

- partial attribute release with selective disclosure through dedicated access rights coded in the cv-certificate,
- unlinkability of transactions by using different values in each protocol run and no ICC-individual private and public keys in CAv2,
- attribute authenticity and eavesdropping protection through end-to-end encryption and authenticity and integrity protection,
- cloning protection through Chip Authentication sub-protocol, and
- together with PACE protocol, a user-centric system due to required user consent.

Furthermore, the protocol provides some additional properties not defined in 6.4:

- unlinkability of attribute holder across domains, ensured by the pseudonym generated via RI or PSA protocols;
- unlinkability of attribute requests of requesting SP towards an attribute provider as the link of the specific attribute to the SP is performed internally by the ICC;
- pseudonymous transferable proof of attribute possession with the PSC sub-protocol.

7.3.5.2 Protocol sequence

In the first phase, the protocol requires password verification based on PACE (see Steps 4 to 15 in Figure 8). Therefore, the IFD provides a password to the ICC for verification. The IFD is either in possession of a suitable password or it requests the user to provide a password. A secure messaging session is established upon execution of PACE protocol. If Enhanced Role Authentication is supported by the ICC, the ICC shall store the PACE session context #1 (ISO/IEC 7816-8).

In the second phase, the ICC and the service provider, i.e. the respective eID-Server, mutually authenticate as genuine by performing the EACv2 protocol (see Steps 16 to 23 in Figure 8). A secure messaging session is established between the ICC and the service provider upon execution of EACv2 protocol. The ICC shall grant access rights according to the effective authorization. If Enhanced Role Authentication is to be used, the service provider shall instruct the ICC to store session information as Session Context #2.

In the third phase, the authenticated terminal may optionally select and use the application(s) according to the effective authorization of the service provider.

The protocol continues if additional attributes not yet stored in the ICC are requested by the service provider. In this fourth phase, the service provider, having effective authorization to write attribute requests, writes an attribute request to the ICC (see Step 25 in Figure 8). If specific attributes are requested, the ICC shall make an internal link between the request and the domain-specific identifier of the requesting service provider (see Step 24 in Figure 8) and shall restrict read access to this attribute request to authenticated attribute providers. The service provider shall initiate the restoration of the PACE Session Context #1 stored by the ICC. The ICC SHALL store the Chip Authentication Session Context #2 prior to restarting Secure Messaging (see Steps 26-27 in Figure 8).

In the fifth phase, the ICC and the attribute provider mutually authenticate as genuine by applying the EACv2 protocol (see Steps 29 to 35 in Figure 8). The ICC shall grant access rights according to the effective authorization. The authenticated attribute provider reads the stored attribute request from the ICC and writes the resulting attributes to the ICC (see Steps 36 to 38 in Figure 8). The ICC shall restrict read access to the stored specific attributes to the service provider authenticated during the preceding EACv2 protocol. It shall restrict read access to generic attributes to service providers with the required authorization.

In the sixth phase, the attribute provider shall initiate the restoration of the PACE Session Context #1. Subsequently, the IFD shall initiate restoration of the Chip Authentication Session Context #2 (see Steps 39 to 42 in Figure 8).

In the seventh and last phase, the service provider may read the credentials according to the effective authorization and grant the user access to the requested service (see Steps 43 to 45 in Figure 8).

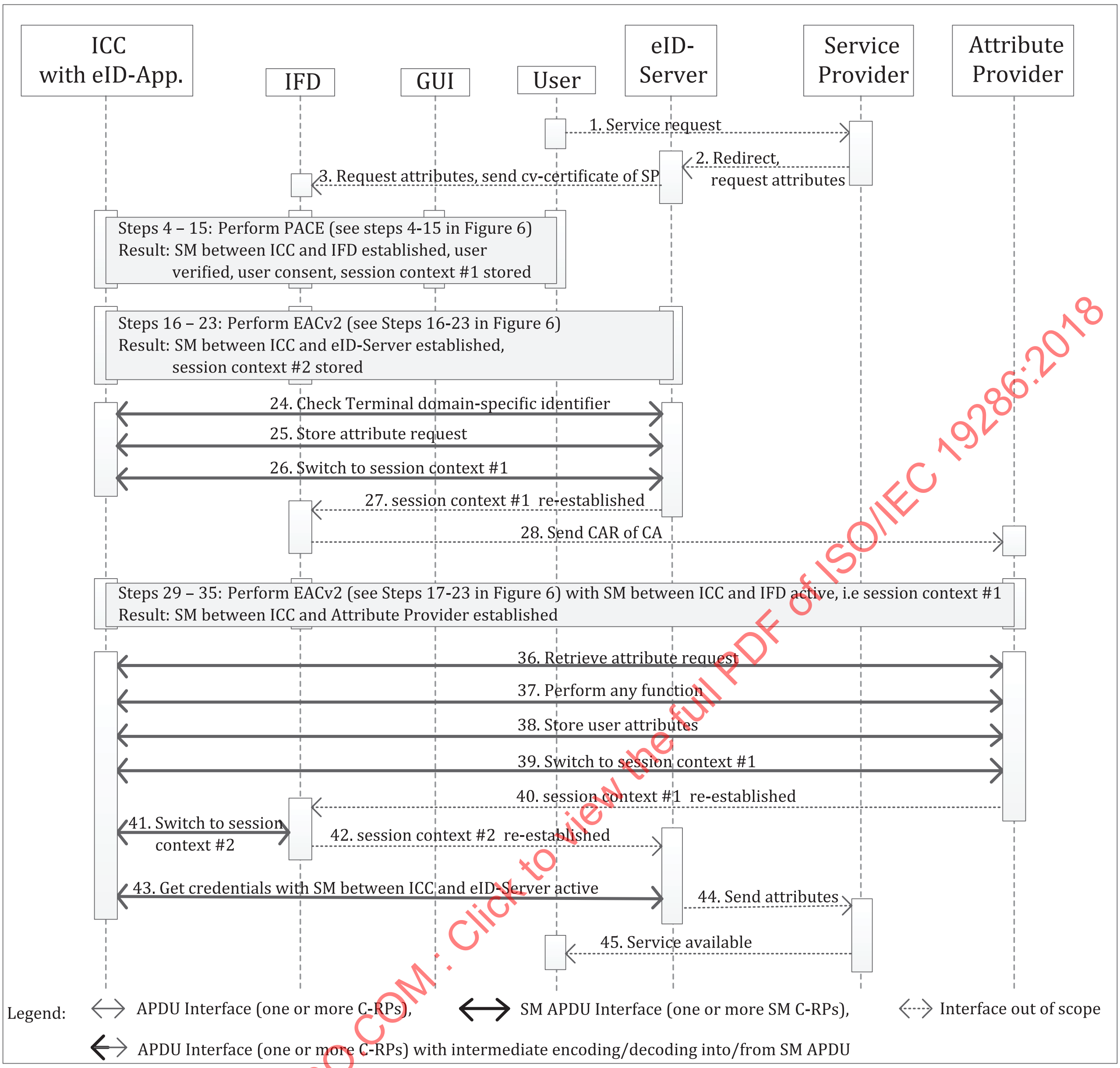
ICC with eID-App.
IFD
GUI
User
eID-Server
Service Provider
Attribute Provider
1. Service request
2. Redirect, request attributes
3. Request attributes, send cv-certificate of SP
Steps 4 – 15: Perform PACE (see steps 4-15 in Figure 6)
Result: SM between ICC and IFD established, user verified, user consent, session context #1 stored
Steps 16 – 23: Perform EACv2 (see Steps 16-23 in Figure 6)
Result: SM between ICC and eID-Server established, session context #2 stored
24. Check Terminal domain-specific identifier
25. Store attribute request
26. Switch to session context #1
27. session context #1 re-established
28. Send CAR of CA
Steps 29 – 35: Perform EACv2 (see Steps 17-23 in Figure 6) with SM between ICC and IFD active, i.e session context #1
Result: SM between ICC and Attribute Provider established
36. Retrieve attribute request
37. Perform any function
38. Store user attributes
39. Switch to session context #1
40. session context #1 re-established
41. Switch to session context #2
42. session context #2 re-established
43. Get credentials with SM between ICC and eID-Server active
44. Send attributes
45. Service available
Legend:
APDU Interface (one or more C-RPs),
SM APDU Interface (one or more SM C-RPs),
Interface out of scope
APDU Interface (one or more C-RPs) with intermediate encoding/decoding into/from SM APDU

Figure 8 — Enhanced Role Authentication (ERA)

7.3.5.3 Data types and relying entities

Table 18 shows data types and relying entities of Enhanced Role Authentication.

Table 18 — Data flow of ERA

Step	Sending entity	Receiving entity	Data type	Description
24	eID-Server	ICC	SP data	Check the terminal domain-specific identifier against the identifiers obtained from cv-certificate in Step 17
25	eID-Server	ICC	SP data	Store the attribute request linked with the terminal domain-specific identifier in Step 24
26	eID-Server	ICC	SP data	Switch to Session Context #1, PACE SM resumes between the ICC and the IFD
27	eID-Server	IFD	n/a	eID-Server informs IFD about change of session context

Table 18 *(continued)*

Step	Sending entity	Receiving entity	Data type	Description
28	IFD	AtP	ICC data	Certification Authority Reference (CAR) of ICC Root-CA is sent by IFD to the attribute provider to initiate the EACv2 protocol that follows
29-35				Mutual authentication according EACv2 is performed between attribute provider and the ICC
36	ICC	AtP	ICC data	Retrieve the attribute request stored by the eID-Server in Step 25
37	AtP / ICC	ICC / AtP	AtP data	Attribute provider can now carry out any necessary process to generate the user attributes requested by the SP (operations are application–dependent here)
38	AtP	ICC	AtP data	Once generated, user attributes are securely stored on-board the ICC
39	AtP	ICC	AtP data	Switch to Session Context #1, PACE SM resumes between the ICC and the IFD
40	AtP	IFD	n/a	AdP informs IFD about change of session context
41	IFD	ICC	n/a	Switch to Session Context #2, the switch is active for the next APDU command CA2 SM resumes (on the ground of parameters stored in context#2) between ICC and the eID-Server
42	IFD	eID-Server	n/a	IFD informs eID-Server about change of session context
43	ICC	eID-Server	User data	eID-Server can now securely read the user attributes available in the ICC and delivered by the attribute provider
44	eID-Server	SP	User data	Transmit the user attributes read from the ICC to the service provider
45	SP	User	SP data	Service is available for user

7.3.5.4 C-RP description

7.3.5.4.1 C-RP sequence

Table 19 shows the C-RP sequence of Enhanced Role Authentication.

Table 19 — C-RP sequence of ERA

Step	Command – INS	P1-P2	Command data field, Le field	Response data field, SW1-SW2
PACE has been performed (see 7.2.3.1), SM established, CAR_1 of root CA is available at IFD, session context #1 stored				
EACv2 has been performed (see 7.3.3.4.2), SM established, session context #2 stored				
24	PUO UO – ‘14’	‘00 80’	{‘7F21’ – L – certificate content template '73' – L – discretionary data template '80' – L – Hash of sector public key} C-RP description (see 7.3.5.4.2)	Data field absent
25	PUT DATA – ‘DA’	‘FF 01’	{‘53’ – L – attribute request} Write attribute request onto the ICC (see 7.3.5.4.2)	Data field absent
26	MSE SET AT – ‘22’	‘01 A4’	{‘E1’ – L – certificate content template '81' – L – session context identifier } Restore security session according to ISO/IEC 7816-8	Data field absent

Table 19 *(continued)*

Step	Command – INS	P1-P2	Command data field, Le field	Response data field, SW1-SW2
EACv2 is to be performed, SM established				
36	GET DATA – ‘CA’	‘FF 01'	absent Get attribute request from ICC	{'53' – L – Attribute Request} SW1-SW2: warning (see ISO/IEC 7816-4), if no attribute request available
37			Perform any action (e.g. provisioning and/or refreshing tokens) (see 7.3.5.4.4)	
38	PUT DATA – ‘DA’	‘00 FF’	{‘53’ – L – specific attribute} – {‘53’ – L – specific attribute} – ... Write a list of specific attributes according to attribute request onto the ICC (see 7.3.5.4.5)	Data field absent
39	MSE SET AT – ‘22’	‘01 A4’	{‘E1’ – L – certificate content template ‘81' – L – session context identifier } Restore security session according to ISO/IEC 7816-8	Data field absent
41	MSE SET AT – ‘22’	‘01 A4’	{‘E1’ – L – certificate content template ‘81’ – L – session context identifier } Restore security session according to ISO/IEC 7816-8	Data field absent
43a	GET DATA – ‘CA’	‘00 FF’	absent Get credentials, i.e. user-specific attributes, from ICC (see 7.3.5.4.6)	See 7.3.5.4.6 SW1-SW2: warning (see ISO/IEC 7816-4), if no attributes available
43b	DELETE DATA – ‘EE’	‘00 00’	absent Delete specific attributes stored in ICC and linked to the terminal domain-specific identifier (see C-RP description in ISO/IEC 7816-9)	Data field absent SW1-SW2: see ISO/IEC 7816-9
Service provider may perform any other protocol in accordance to effective access rights				

7.3.5.4.2 Step 24 — Terminal domain-specific identifier

In contrast to domain-specific identifiers (pseudonyms) computed by the ICC and used to identify a user within a specific sector, the "Terminal domain-specific identifier" is to be interpreted the other way round, i.e. the ICC shall identify a terminal as part of a specific sector. The ICC shall check if the identifier, i.e. hash of sector public key transmitted by PUO command, is included in the certificate extension of the already verified card-verifiable certificate during Terminal Authentication.

7.3.5.4.3 Step 25 — Write attribute request

For Enhanced Role Authentication, attribute requests stored on the ICC shall be of the following type (see ASN.1 definition).

```
RequestInfos ::= SET OF RequestInfo

RequestInfo ::= SEQUENCE {
        requestType OBJECT IDENTIFIER,
        [0] requiredData ANY DEFINED BY requestType,
        [1] optionalData ANY DEFINED BY requestType OPTIONAL
         }
```

The command data field of the PUT DATA command shall encode only one object of type RequestInfos.

7.3.5.4.4 Step 37 — Authentication token management

Once secure messaging is established at Step 32 (see Figure 8), and the authentication tokens are authorized to be written on the ICC, the server either delivers or refreshes such tokens onto the ICC. The tokens already on-board the ICC may be renewed (e.g. in case their lifespan is elapsed). See A.9 for an example use case.

7.3.5.4.5 Step 38 — Write user attributes

The following ASN.1 structure shall be used to store Attributes for Enhanced Role Authentication on the ICC.

```
Attribute ::= SEQUENCE {
        attributeType OBJECT IDENTIFIER,
        [0] requiredData ANY DEFINED BY attributeType
        [1] optionalData ANY DEFINED BY attributeType
        }
```

7.3.5.4.6 Step 43a — Get user attributes

The GET DATA command responds with a sequence of specific attributes encapsulated in a discretionary data template according to the domain-specific identifier of the terminal computed in Step 24 (see 7.3.5.4.2). If the effective access right of the service provider allows the access to all terminal sectors, a sequence of specific attributes linked to the respective domain-specific identifier and encapsulated in a discretionary data template is transmitted in the response data. Table 20 shows the structure of response data of GET DATA command.

Table 20 — Structure of response data

‘73’	Discretionary data template	
	‘53’	Specific attribute
	…	Further specific attributes (optional)
‘73’	Discretionary data template (conditional)	
	‘80’	Domain specific identifier
	‘53’	Specific attribute
	…	Further specific attributes (optional)
‘73’	Further discretionary data templates with TAG '80' (conditional)	
	‘80’	Domain specific identifier
	…	
…		

7.3.5.5 Protocol-dependent descriptions

7.3.5.5.1 User and service provider authentication

The relative authorization of the service provider is encoded in the cv-certificate (see ISO/IEC 7816-8) in the CHAT data object (see Tables 21 and 22). If bit 6 ("Access to all terminal sectors") in Table 21 is not set, the ICC shall restrict read and delete access to the specific attributes tied to the domain-specific identifier of the service provider. If bit 6 is set, the ICC shall grant read and delete access to all attributes, i.e. generic and specific attributes.

Table 21 — Certificate holder authorization for ERA specific attributes

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x	x	x	x	x	x	x	**Access rights (attribute)**
x	—	—	—	—	—	—	—	RFU
—	1	—	—	—	—	—	—	PSC
—	—	1	—	—	—	—	—	Access to all terminal sectors
—	—	—	1	—	—	—	—	Delete specific attributes
—	—	—	—	1	—	—	—	Write specific attribute
—	—	—	—	—	1	—	—	Read specific attribute
—	—	—	—	—	—	1	—	Write attribute request
—	—	—	—	—	—	—	1	Read attribute request
— Any other value is RFU.								

Table 22 — Certificate holder authorization for ERA generic attributes

b32	b31	b30	...	b11	b10	b9	...	b2	b1	Meaning
x	x	x	...	x	x	x	...	x	x	**Access rights (attribute)**
1	—	—	...	—	—	—	...	—	—	Erase generic attribute 8
—	1	—	...	—	—	—	...	—	—	Write generic attribute 8
—	—	1	...	—	—	—	...	—	—	Read generic attribute 8
—	—	—	...	—	—	—	...	—	—	...
—	—	—	...	1	—	—	...	—	—	Erase generic attribute 1
—	—	—	...	—	1	—	...	—	—	Write generic attribute 1
—	—	—	...	—	—	1	...	—	—	Read generic attribute 1
—	—	—	...	—	—	x	...	x	—	RFU
—	—	—	...	—	—	—	...	—	1	PSC
— Any other value is RFU.										

7.3.5.5.2 Attribute provider authentication

The EACv2 device authentication protocol between the ICC and the attribute provider is to be performed. Attribute providers shall have access right “Read attribute requests” coded in the CHAT object of the card-verifiable certificate. Attribute providers with authorization to provide specific attributes shall have access right “Write specific attribute” set.

7.3.6 Device authentication protocol OPACITY Full Secrecy

7.3.6.1 General protocol description

The OPACITY Full Secrecy protocol provides mutual authentication between the ICC and IFD. This protocol is based on Bilateral Key Confirmation Scheme[11]. The session keys are derived from the IFD static key and ephemeral key and the ICC static key and ephemeral key. The successful completion of this protocol provides each entity an assurance that they generated correct session keys and that other entity actively participated in the key generation process.

The protocol provides two modes of operations, namely, Persistent Binding and Non-Persistent Binding. In the Persistent Binding mode, the IFD and ICC generate and store shared secret for the next session to facilitate faster key establishment. In the Non-Persistent Binding mode, the IFD and the ICC generate ephemeral keys each time to generate session keys.

The protocol provides

— unlinkability by using different ephemeral values from both the ICC and IFD. Since the data is encrypted from the very first interaction between the ICC and the IFD, anybody able to observe the interaction will not be able to link two different sessions,

— attribute authenticity and eavesdropping protection through end-to-end encryption and integrity protection from the very first interaction, and

— cloning protection if the private static key material is stored in a hardware security module and is never exposed in the communication between the ICC and IFD.

NOTE A security analysis of the OPACITY Full Secrecy protocol can be found in Reference [28].

[11] See http://dx.doi.org/10.6028/NIST.SP.800-56Ar2.

7.3.6.2 Protocol sequence

Figure 9 shows the protocol sequence of device authentication protocol OPACITY FS.

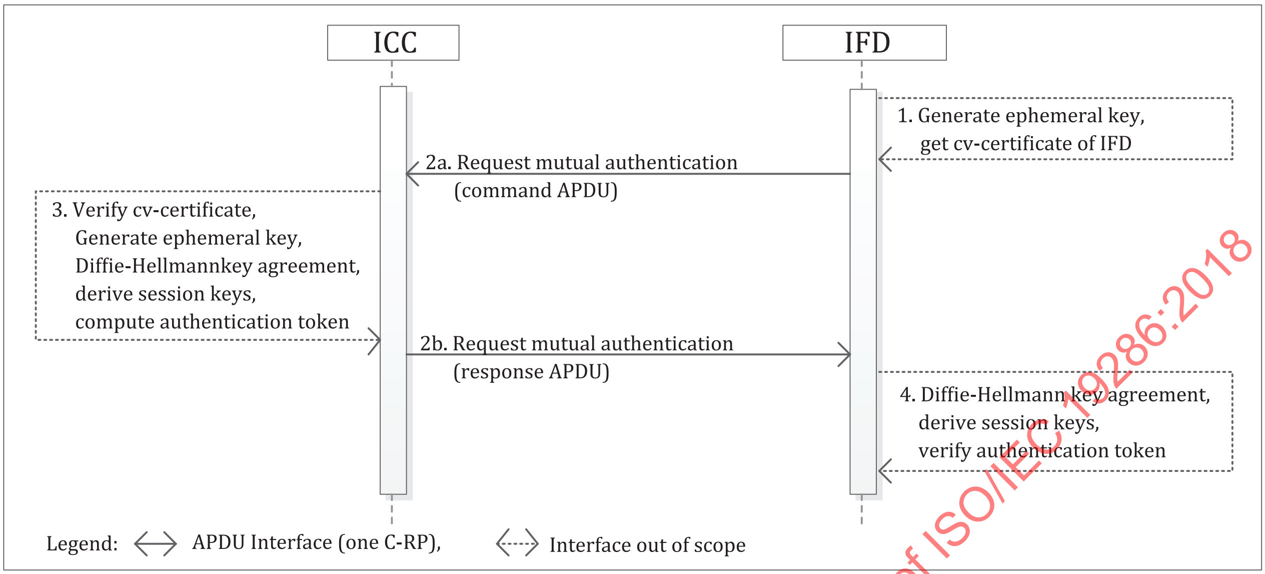
ICC
IFD
1. Generate ephemeral key,
get cv-certificate of IFD
2a. Request mutual authentication
(command APDU)
3. Verify cv-certificate,
Generate ephemeral key,
Diffie-Hellmannkey agreement,
derive session keys,
compute authentication token
2b. Request mutual authentication
(response APDU)
4. Diffie-Hellmann key agreement,
derive session keys,
verify authentication token
Legend:
APDU Interface (one C-RP),
Interface out of scope

Figure 9 — Device authentication with OPACITY FS

7.3.6.3 Data types and relying entities

Table 23 shows data types and relying parties of device authentication protocol OPACITY FS.

Table 23 — Data flow of OPACITY FS device authentication protocol

Step	Sending party	Receiving party	Data type	Description
1	IFD	IFD	IFD data	IFD generates ephemeral public/private key pair. This may be computed in advance. IFD receives cv-certificate.
2a	IFD	ICC	IFD data	IFD provides the ephemeral public key and its cv-certificate to the ICC and it requests the ICC to initiate Diffie-Hellmann key agreement protocol and establish session keys.
3	ICC	ICC	ICC Data IFD Data	ICC verifies cv-certificate performs Diffie-Hellmann key agreement, derives secure messaging keys and computes authentication token.
2b	ICC	IFD	ICC data	ICC sends cv-certificate holding static public key or ICC nonce and authentication token
4	IFD	IFD	IFD Data ICC Data	IFD verifies cv-certificate, performs Diffie-Hellmann key agreement, derives secure messaging keys and verifies authentication token.

7.3.6.4 C-RP description

7.3.6.4.1 C-RP sequence

Table 24 shows the C-RP sequence of OPACITY Full Secrecy device authentication protocol.

Table 24 — C-RP sequence of OPACITY Full Secrecy device authentication protocol

Step	Command – INS	P1-P2	Command data field, Le data field	Response data field, SW1-SW2
IFD ephemeral key generated, IFD cv-certificate received				
2a/b	GENERAL AUTHENTICATE – ‘86’	‘xx yy’	‘xx’ – cypher suite ‘yy’ – private key reference or ‘00’ Data field is concatenation of { CB_H || C_H || Q_{eH} } See 7.3.6.4.2 for further information	Data field is concatenation of { CB_{ICC} || $OpaqueData_{ICC}$ || $AuthCryptogram_{ICC}$ || $OTID_{ICC}$ } See 7.3.6.4.2 for further information SW1-SW2 according to ISO/IEC 7816-8
Secure messaging established				

7.3.6.4.2 Step 2a/b — Request mutual authentication

The following data are to be transmitted in the data field.

CB_H	is the encoding of Persistent vs Non-Persistent Binding mode.
C_H	is the IFD cv-certificate encoding static public key and issued by the trust anchor of the ICC.
Q_{eH}	IFD ephemeral public key.
CB_{ICC}	is the encoding of Persistent Binding vs Non-Persistent Binding.
$OpaqueData_{ICC}$	is either encrypted ICC cv-certificate or ICC nonce, depending on the mode of operation.
$AuthCryptogram_{ICC}$	is the result of key confirmation computation.
$OTID_{ICC}$	is the ICC anonymous identifier, valid one time.

7.3.7 Device authentication protocol OPACITY BLINDED

7.3.7.1 General protocol description

This protocol establishes shared session keys between the IFD and the ICC where only the ICC is authenticated. This protocol is based on the blinded Diffie-Hellmann key agreement scheme described in Mechanism 13 of Amendment 1 of Reference [7]. The session keys are derived from the IFD ephemeral key and the ICC static key. The protocol is designed to provide privacy and tracking protection against passive eavesdropping.

The protocol provides

— unlinkability by using different ephemeral values from the IFD for each run and by encrypting the ICC identifier using the IFD ephemeral key and ICC nonce. Since the data is encrypted from the very first ICC response, anybody able to observe the interaction will not be able to identify the ICC and therefore, will not be able link two different sessions,

— attribute authenticity and eavesdropping protection through end-to-end encryption and integrity protection from the very first interaction, and

— cloning protection since the private static key material is stored in the ICC and is never exposed in the communication between the ICC and IFD.

NOTE A security analysis of the blinded Diffie-Hellmann protocol can be found in Reference [1].

7.3.7.2 Protocol sequence

Figure 10 shows the protocol sequence of device authentication protocol OPACITY BLINDED.

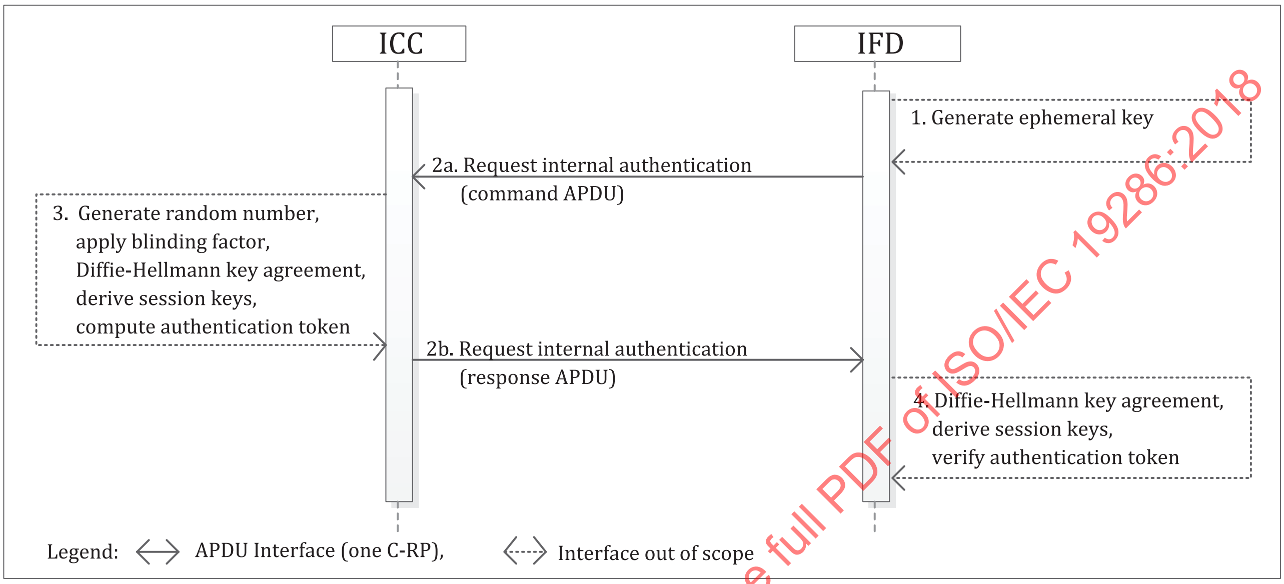
ICC
IFD
1. Generate ephemeral key
2a. Request internal authentication (command APDU)
3. Generate random number, apply blinding factor, Diffie-Hellmann key agreement, derive session keys, compute authentication token
2b. Request internal authentication (response APDU)
4. Diffie-Hellmann key agreement, derive session keys, verify authentication token
Legend: APDU Interface (one C-RP), Interface out of scope

Figure 10 — ICC device authentication with OPACITY BLINDED

7.3.7.3 Data types and relying entities

Table 25 shows data types and relying parties of device authentication protocol OPACITY BLINDED.

Table 25 — Data flow of OPACITY BLINDED device authentication protocol

Step	Sending party	Receiving party	Data type	Description
1	IFD	IFD	IFD data	IFD generates ephemeral public/private key pair. This may be computed in advance.
2a	IFD	ICC	IFD data	IFD provides the ephemeral public key and requests the ICC to initiate Diffie-Hellmann key agreement protocol and establish session keys.
3	ICC	ICC	ICC Data IFD Data	ICC generates random number, applies blinding factor and performs Diffie-Hellmann key agreement, derives secure messaging keys and computes authentication token.
2b	ICC	IFD	ICC data	ICC sends anonymized identifier, encrypted cv-certificate holding static public key and authentication token
4	IFD	IFD	IFD Data ICC Data	IFD verifies cv-certificate, performs Diffie-Hellmann key agreement, derives secure messaging keys and verifies authentication token.

7.3.7.4 C-RP description

7.3.7.4.1 C-RP sequence

Table 26 shows the C-RP sequence of OPACITY BLINDED device authentication protocol.

Table 26 — C-RP sequence of OPACITY BLINDED device authentication protocol

Step	Command – INS	P1-P2	Command data field, Le data field	Response data field, SW1-SW2
IFD ephemeral key generated				
2a/b	GENERAL AUTHENTICATE – ‘86’	‘xx yy’	‘xx’ – cypher suite ‘yy’ – private key reference or ‘00’ Data field is concatenation of { CB_H || ID_H || Q_{eH} } See 7.3.7.4.2 for further information	Data field is concatenation of { CB_{ICC} || $OpaqueData_{ICC}$ || $AuthCryptogram_{ICC}$ || $BlindedPub_{ICC}$ } See 7.3.7.4.2 for further information SW1-SW2 according to ISO/IEC 7816-8
Secure messaging established				

7.3.7.4.2 Step 2a/b — Request internal authentication

The following data are to be transmitted in the data field.

CB_H is the encoding of Persistent vs Non-Persistent Binding mode. CB_H shall be set to ‘00’.

ID_H is the IFD identifier.

Q_{eH} is the IFD ephemeral public key.

CB_{ICC} is the encoding of Persistent Binding vs Non-Persistent Binding. CB_{ICC} shall be set to‘00’.

$OpaqueData_{ICC}$ encrypted ICC cv-certificate.

$AuthCryptogram_{ICC}$ is the result of key confirmation computation.

$BlindedPub_{ICC}$ is ICC anonymized identifier. This is the random number multiplied by the ICC public key.

7.4 Attribute verification mechanisms with COMPARE command

7.4.1 Purpose of attribute verification mechanism

Attribute verification mechanisms provide for retrieving attribute statements over user attributes from the ICC and hence, allow for the verification of user attributes without reading attributes from the ICC. A typical example is age verification by verifying if the age of a certain user is greater or smaller than a given date without revealing the exact date of birth. Additionally, other attribute data, such as name and/or gender of the user, is not revealed during the processing.

7.4.2 General

The COMPARE command initiates a comparison of "comparison data" with non-volatile "reference data" stored in the ICC. The comparison data is provided by the service provider either in the COMPARE command or by an operation prior to the comparison (e.g. as “auxiliary data” as part of the EACv2 protocol). Typical examples are age verification, document validity verification, place of residence or nationality verification. Further data elements that might be target of privacy protection are given in Table 27.

Table 27 — Data elements for privacy protection according to ISO/IEC 7816-6

TAG	Data element
‘5F2C’	Cardholder nationality
‘5F2B’	Date of birth
‘5F2D’	Language preferences
‘5F35’	Gender
‘5F26’	Card effective date
‘5F25’	Application effective date

NOTE As with embedding the comparison data in the command data field of the COMPARE command, the command could be sent multiple times with different comparison data. This would allow for an approximation of the reference data and violate privacy requirements. For example, the service provider could evaluate the actual date of birth, the reference data, by iterative verification attempts with different calendar dates, the comparison data. If the service provider is allowed to send the compare command only once, then such an attempt can be avoided.

To respect privacy requirements, the execution of the COMPARE command shall follow appropriate access conditions and shall be embedded in a privacy-enhancing protocol. This clause specifies those protocols. The COMPARE command responds with a successful or failed comparison, which results in the retrieving of an attribute statement over the particular attribute, i.e. the reference data. Table 28 gives possible status words of the COMPARE command.

Table 28 — Return codes of COMPARE command

SW1-SW2	Meaning
‘9000’	Comparison successful
‘6282’	End of file or record
‘6300’	Comparison failed, no comparison counter set
‘63xx’	Comparison failed, 'xx' number of remaining comparison attempts
‘6340’	Comparison failed
‘6982’	Security condition not satisfied
‘6A88’	Comparison or reference data not found

7.4.3 Data comparison with external authentication function

7.4.3.1 General protocol description

Data verification with the COMPARE command shall be secured by secure messaging and a mutual authentication between the IFD and the ICC in order to restrict the access to attribute statements to authorized IFDs only and in order to enable the IFD to proof the integrity and authenticity of the received response, i.e. the attribute statement.

Mechanisms to get user consent for the release of attribute statements shall be applied. Moreover, mechanisms, i.e. processing counters, restricting the unlimited processing of COMPARE commands shall be applied to avoid leakage of the user attributes by recursively questioning attribute statements.

EXAMPLE In the age verification example, by sending a hundred times "is the user older than …" with a decremented day date, an IFD may eventually get the exact date of birth.

In Step 1, an application, i.e. an ADF or DF, is selected that features the user attribute in any kind of object. In optional Step 2, the environment for comparison is set. From Step 3 to Step 6, a key for SM and a mutual authentication is established between the ICC and the IFD with four GENERAL AUTHENTICATE commands. In optional Step 7, the target data is selected. Finally in Step 8, the verification with COMPARE command with the requested attribute statement as response is processed on the ICC.

Data comparison by means of COMPARE command provides partial attribute release, according to 6.4, as only attribute statements are sent to the requesting party.

7.4.3.2 Protocol sequence

Figure 11 shows the protocol sequence of data comparison with External Authentication function.

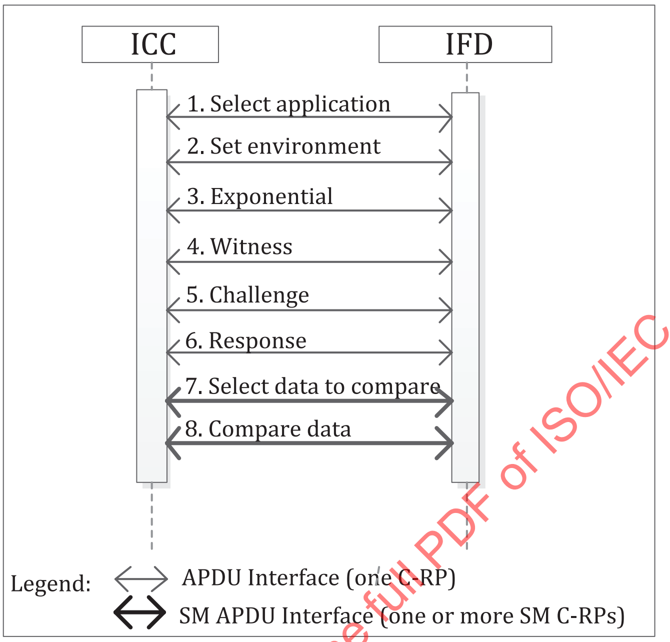

Figure 11 — Data comparison with authentication protocol

7.4.3.3 C-RP description

Table 29 shows the C-RP description of data comparison with External Authentication function.

Table 29 — C-RP sequence of data comparison with External Authentication function

Step	Command – INS	P1-P2	Command data field, Le data field	Response data field, SW1-SW2
Key agreement according to ISO/IEC 7816-4:2013, C.1.4 has been performed (Steps 3 to 6) Optionally, comparison data have been selected (Step 7)				
8	COMPARE – '33'	'02 03'	{0E 5C 0C 60 04 5C 02 {5F 2B 53 04 19 94 09 01} date of birth} Example age verification: comparison whether the birthday is less than '19940901'	Data field absent SW1-SW2 (see Table 28)
.....				

7.4.4 Auxiliary data comparison with EACv2 protocol

7.4.4.1 General protocol description

The comparison data shall be sent as auxiliary data as part of the EACv2 protocol. For instance, the eID-Server may send an age threshold value as comparison data (e.g. the actual calendar date minus 18 years for verification of "older than 18") as part of the AUX data in the EACv2 protocol. The ICC performs a "comparison data greater than date of birth" comparison and shall return acceptance or refusal without having to reveal the actual age of the card holder.

User control is achieved by performing the PACE protocol prior to EACv2 protocol and by setting the appropriate access right in the CHAT objects.

Data comparison by means of COMPARE command and EACv2 together with PACE protocol provides partial attribute release, according to 6.4, as only attribute statements are sent to the requesting entity.

7.4.4.2 Protocol sequence

Figure 12 shows the protocol sequence of auxiliary data comparison with EACv2 protocol.

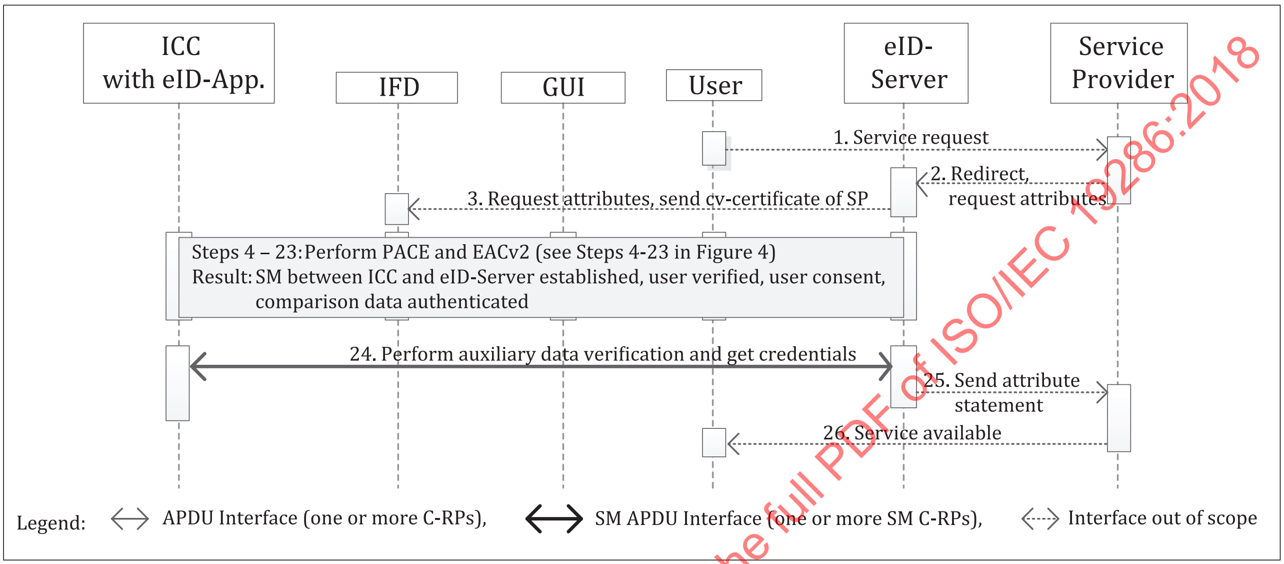
ICC with eID-App.
IFD
GUI
User
eID-Server
Service Provider
1. Service request
2. Redirect, request attributes
3. Request attributes, send cv-certificate of SP
Steps 4 – 23:Perform PACE and EACv2 (see Steps 4-23 in Figure 4)
Result: SM between ICC and eID-Server established, user verified, user consent, comparison data authenticated
24. Perform auxiliary data verification and get credentials
25. Send attribute statement
26. Service available
Legend:
APDU Interface (one or more C-RPs),
SM APDU Interface (one or more SM C-RPs),
Interface out of scope

Figure 12 — Auxiliary data comparison protocol with EACv2

7.4.4.3 Data types and relying entities

For data types and relying entities, see Table 10, whereas in Step 24, no user attributes but attribute statements are received from the ICC, i.e. comparison successful or failed.

7.4.4.4 C-RP description

7.4.4.4.1 C-RP sequence

Table 30 shows the C-RP sequence of auxiliary data comparison with EACv2 protocol.

Table 30 — C-RP sequence of auxiliary data comparison protocol with EACv2

Step	Command – INS	P1-P2	Command data field, Le field	Response data field, SW1-SW2
EAC has been performed (see 7.3.3), SM established, auxiliary data have been authenticated				
24	COMPARE – ‘33’	‘00 00’	{‘06’ – L – OID defining comparison and reference to comparison data and to reference data} – {‘60’ – ‘00’ – empty, defined by OID} Compare data imported during EAC with stored data in ICC (see 7.4.4.4.2)	Data field absent SW1-SW2 (see Table 28)
Apply COMPARE command in accordance to authenticated auxiliary data in EAC protocol				

7.4.4.4.2 Step 24 — Auxiliary data comparison

The comparison data have been already imported into the ICC as auxiliary data in DO‘67’ in Step 20a of EACv2 protocol (see Figure 6 and Table 11) and have been authenticated with successful Terminal Authentication in Step 20c. Both the comparison data and reference data are known to the ICC. The OID transmitted in the data field of the COMPARE command is a reference to the comparison data and reference data. The ICC shall use the corresponding data and the comparison in accordance to the OID. If two or more comparison data of the same type, i.e. with identical OID, have been imported into the ICC, the ICC shall use the last imported comparison data for comparison.

7.5 Domain-specific identifier mechanisms

7.5.1 Purpose of domain-specific identifier mechanisms

Domain-specific identifiers provide a unique identifier within a certain domain but not across different domains. This mechanism prevents use of one unique ICC identifier across different applications, domains or sectors. Thus, this mechanism prevents user profiling.

7.5.2 Domain-specific identifier based on Restricted Identification

7.5.2.1 General protocol description

Restricted Identification provides a domain-specific identifier for the ICC with the following properties:

- within each sector, the sector-specific identifier of every ICC is unique;
- across any two sectors, it is computationally infeasible to link the domain-specific identifiers of any token. Depending on the generation of the domains, a trusted third-party may or may not be able to link domain-specific identifiers between domains.

The private key of the ICC used for Restricted Identification protocol shall be securely stored within the ICC and should be known by the ICC only.

A Terminal Authentication shall be performed to ensure, that the service provider is authentic and the presented terminal certificate belongs to the terminal. A Chip Authentication shall also be performed so the service provider can verify the authenticity of the ICC and attempt to alter or masquerade domain-specific identifiers are prevented.

The domain of the service provider and the hash value of the public key intended to be used for Restricted Identification have to be embedded in the terminal's cv-certificate so an attacker cannot switch the sector information to obtain the domain-specific identifier for a different domain.

The Restricted Identification protocol fulfills partial attribute release property and domain-specific identifier property according to 6.4.

7.5.2.2 Protocol sequence

Figure 13 shows the protocol sequence of Restricted Identification based on Diffie-Hellman.

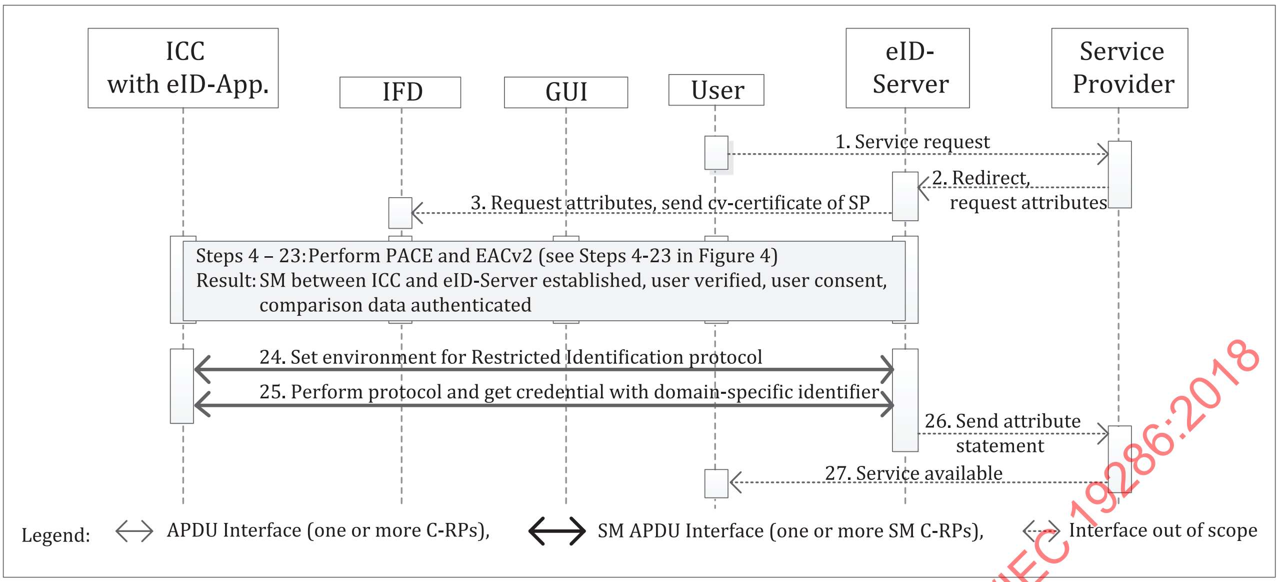
ICC with eID-App.
IFD
GUI
User
eID-Server
Service Provider
1. Service request
2. Redirect, request attributes
3. Request attributes, send cv-certificate of SP
Steps 4 – 23: Perform PACE and EACv2 (see Steps 4-23 in Figure 4)
Result: SM between ICC and eID-Server established, user verified, user consent, comparison data authenticated
24. Set environment for Restricted Identification protocol
25. Perform protocol and get credential with domain-specific identifier
26. Send attribute statement
27. Service available
Legend:
APDU Interface (one or more C-RPs),
SM APDU Interface (one or more SM C-RPs),
Interface out of scope

Figure 13 — Restricted Identification protocol based on Diffie-Hellman

7.5.2.3 Data types and relying entities

For data types and relying entities, see Table 10, whereas in Step 25, no user attributes but attribute statements are received from the ICC, i.e. the domain-specific identifier.

NOTE The ICC-individual private key used to perform the protocol can be considered as "user attribute". The computed sector-identifier with the help of the terminal public key could be considered as the statement over the user attribute.

7.5.2.4 C-RP description

7.5.2.4.1 C-RP sequence

In the first step the service provider, i.e. the eID-Server, has to choose the ICC's secret key by sending the correct reference of the private key in an MSE SET AT command. The service provider presents its dedicated public key together with the corresponding domain parameters by applying the GENERAL AUTHENTICATE command in the second step. The ICC evaluates the common secret value of a Diffie-Hellman key agreement and returns in the response APDU the hash value of the x-coordinate of the result to the IFD. Table 31 shows the C-RP sequence of Restricted Identification based on Diffie-Hellman.

Table 31 — C-RP sequence of Restricted Identification protocol

Step	Command – INS	P1-P2	Command data field, Le field	Response data field, SW1-SW2
EAC has been performed (see 7.3.3), SM established, domain parameters have been authenticated by ICC				
24	MSE SET AT – ‘22’	‘41 A4’	{‘80’ – L – OID of RI-protocol} – {‘84’ – L – reference of private key of ICC PrK.ICC.RI} Set key reference of ICC private key for Restricted Authentication	Data field absent
25	GENERAL AUTHENTICATE –‘86’	‘00 00’	{‘7C’ – L – dynamic authentication data {‘AX’ – L – public key data template, see 7.5.2.4.2 } {‘‘6’ – L – OID of public key} – {public key data objects according to OID}}} Send authenticated public key including domain parameter	{‘7C’ – L – {‘8X’ – L – DSID$_{\text{ICC}}$}} Perform DH key exchange and compute domain-specific identifier (see 7.5.2.4.2)
Perform again Restricted Identification protocol according to authenticated public keys				

7.5.2.4.2 Step 25 – DH key agreement

The coding of the DO ‘AX’ is shown in Table 32.

Table 32 — Coding of DO 'AX' in CAPDU

TAG in CAPDU	TAG in cv-certificate	TAG in RAPDU	RI key
‘A0’	‘80’	‘81’	PuK.IFD.RI.1
‘A2’	‘81’	‘83’	PuK.IFD.RI.2

The response data field contains the hash of the shared secret, i.e. the domain-specific identifier, DSID$_{\text{ICC}}$, HASH(x(PrK.ICC.RI · PuK.SP.RI)).

7.5.3 Domain-specific identifier based on pseudonymous signature for authentication

7.5.3.1 General protocol description

The computation of domain-specific identifiers is an optional part of CA-PSA as described in 7.6.2. The "domain" is determined by the terminal by a static public key of the terminal. The protocol Terminal Authentication shall be performed before CA-PSA and the terminals access certificate shall convey a hash value of the terminal's static public key in the certificate extension. If the terminal provides such public key as part of TAv2 and CA-PSA, the ICC shall compute up to two domain-specific identifiers unique for the terminal's static public key. The computation of the domain-specific identifiers depends on the access rights of the ICC's secret keys for computation of pseudonymous signatures.

NOTE A security analysis of the creation of pseudonymous signatures together with the creation of domain-specific identifiers can be found in Reference [21].

7.5.3.2 Protocol sequence

See Figure 6 together with CA-PSA description in Table 35.

7.5.3.3 Data types and relying entities

Table 33 shows data types and relying entities of CA-PSA.

Table 33 — Data flow of domain-specific identifier based on CA-PSA protocol

Step	Sending entity	Receiving entity	Data type	Description
See Table 10 for Steps 1 to 21.				
22	eID-Server	IFD	ICC data SP data	Perform CA-PSA by applying Diffie-Hellman key agreement with exchanged ephemeral public key of ICC and eID-Server and verify pseudonymous signature over ephemeral public key of ICC. Compute domain-specific identifiers based on static public key of eID-Service (on behalf of SP).
23	IFD	ICC	ICC data SP data	Transform APDUs into SM-APDUs.
See Table 10 for Steps 24 to 26.				

7.5.3.4 C-RP description

See C-RP description Steps 23a to 23d for CA-PSA in Table 35.

7.5.4 Domain-specific identifier based on ABC-based signatures

The ABC-based signature computation as part of the ABC protocol given in 7.3.4 includes the computation of either one or two sector-specific pseudonyms if a sector public key of the service provider is provided (see 7.3.4.4.3 for details).

7.6 Pseudonymous signature mechanisms

7.6.1 Purpose of pseudonymous signatures

Pseudonymous signature schemes basically allow the creation of signatures with individual private keys whereas the verification of those signatures is done with a shared public key. Hence, pseudonymous signatures provide means to verify a digital signature without revealing the identity of the signer. The verifier can proof that the signer belongs to a certain group of signers. As a unique identifier of the signer is not released, this mechanism fits data minimization well. Pseudonymous signatures are related to anonymous signatures.

Signature creation and verification algorithms, together with assigned object identifiers, can be found in ISO/IEC 20008-2, ISO/IEC 18370-2 and References [25] and [26].

7.6.2 Chip Authentication based on Pseudonymous Signature for Authentication (CA-PSA)

7.6.2.1 General protocol description

The CA-PSA protocol enhances the Chip Authentication version 2 (CAv2) protocol as part of EACv2 by the creation of a pseudonymous signature by the ICC over the ICC chip authentication ephemeral public key.

The CA-PSA protocol requires the generation of an ephemeral key pair by the ICC. The created ICC ephemeral public key for CA-PSA is signed by the ICC and sent to the eID-Server, which verifies the pseudonymous signature by using the "group manager key" for pseudonymous signatures and hence, validates the ICC ephemeral public key. Passive Authentication is to be performed by the eID-Server to validate the group manager key. Hence, the CA-PSA protocol allows both

— the usage of ICC individual key pair to be used in CA, and

— the usage of ICC-individual private key to be used for the generation of a pseudonymous signature in PSA.

The CA-PSA protocol fulfills the unlinkability and pseudonymous signature as well as cloning protection property according to 6.4. As the CA-PSA protocol is an alternative to CAv2 with no ICC-individual private/public key pair, all properties of EACv2 protocol hold when used in EACv2 protocol.

As the CAv2 protocol described in 7.3.3 requires a static private key securely stored on the ICC and to be used in the Diffie-Hellman key agreement, the eID-Server shall validate the corresponding ICC public key by verifying the electronic signature over the ICC public key, i.e. it has to perform Passive Authentication. To respect privacy in terms of tracking or tracing of the ICC, the static key pair for CAv2 should not be chosen ICC-individually but should be shared by a sufficiently large number of ICCs.

NOTE 1 A security analysis of the creation of pseudonymous signatures together with the creation of domain-specific identifiers can be found in Reference [21].

NOTE 2 In the description of EACv2 in Reference [25], the protocol CA-PSA is also referred to as CA Version 3.

7.6.2.2 Protocol sequence

See Figure 6.

7.6.2.3 Data types and relying entities

Table 34 shows data types and relying entities of domain-specific identifier based on Pseudonymous Signature for Authentication.

Table 34 — Data flow of CA-PSA protocol

Step	Sending entity	Receiving entity	Data type	Description
See Table 10 for Steps 1 to 20.				
21	eID-Server	eID-Server	Issuer data ICC data	Perform Passive Authentication, i.e. verify issuer signature of group manager public key for pseudonymous signature verification including verification of chain of X.509 certificates.
22	eID-Server	IFD	ICC data SP data	Perform CA-PSA by applying Diffie-Hellman key agreement with exchanged ephemeral public keys of ICC and eID-Server, derive SM session keys and verify pseudonymous signature over ephemeral public key of ICC.
23	IFD	ICC	ICC data SP data	Transform APDUs into SM-APDUs.
See Table 10 for Steps 24 to 26.				

7.6.2.4 C-RP description

7.6.2.4.1 C-RP sequence

The C-RP sequence of Chip Authentication of the EACv2 protocol is extended by two more steps 23c and 23d (see Steps 23a and 23b in Table 11). As CA-PSA uses ephemeral keys for Diffie-Hellman key agreement, the command encoding of CAv2 is adapted. The C-RP description of domain-specific identifier based on Pseudonymous Signature for Authentication is shown in Table 35.

Table 35 — C-RP description of CA-PSA protocol

Step	Command – INS	P1-P2	Command data field, Le field	Response data field, SW1-SW2
PACE and Terminal Authentication have been performed (see Steps 18a-d, 20a-c in Table 11)				
23a	MSE SET AT – '22' SM activated	'41 A4'	{'80' – L – OID of CA-PSA protocol} – {'84' – L – reference of private key PrK.ICC.AUT} Set protocol parameter and private key object for Chip Authentication (see 7.6.2.4.2)	Data field absent
23b	GENERAL AUTHENTICATE –'86' SM activated	'00 00'	{'7C' – L – dynamic authentication data {'80' – L – ephemeral public key of SP}} Create ICC ephemeral key for CA-PSA and perform DH key agreement and derive SM keys (see 7.6.2.4.2)	{'7C' – L {'81'–L–PuK.ICC.AUT)} Send ICC ephemeral public key for CA-PSA
23c	MSE SET AT – '22' SM activated	'41 A4'	{'80' – L – OID of PSA protocol} – {'84' – L – reference of private key PrK.ICC.PSA.k} Set protocol parameter and private key for pseudonymous signature creation	Data field absent
23d	GENERAL AUTHENTICATE – '86' SM activated	'0000'	{'7C' – L – dynamic authentication data {'80' – L – SP public key for domain-specific identifier }} Create signature and optionally compute domain-specific identifier(s) (see 7.6.2.4.3)	{'7C' – L – {'82'–L–PuK.ID.Sector1 (cond.) }– {'83'–L–PuK.ID.Sector1 (cond.)} – {'84'–L– (c||s1||s2)}} Pseudonymous signature and conditionally domain-specific identifiers (see 7.6.2.4.3)
Proceed with Steps 24a-b in Table 11				

7.6.2.4.2 Step 23a and 23b — Set private key and perform DH key agreement

In Step 23a, the reference to the private key is sent in a MANAGE SECURITY ENVIRONMENT command to set the private key to be used in CA-PSA. In contrast to CAv2 described in 7.3.3, the CA-PSA protocol requires the ICC to generate an ephemeral key pair. Hence, the referenced private key does not hold any key data in Step 23a. The ephemeral key data are generated in Step 23b.

7.6.2.4.3 Step 23c and 23d — Set key and create pseudonymous signature

In Step 23c, the reference to the private pseudonymous signature key together with an OID setting the specific algorithm is sent in a MANAGE SECURITY ENVIRONMENT command to set the private key and the specific algorithm to be used in CA-PSA. The pseudonymous signature over the generated ephemeral public of the ICC (see Step 23a of CA-PSA in Table 35) and optionally, the domain-specific identifiers are computed in Step 23d in a GENERAL AUTHENTICATE command. The signature algorithm, i.e. the OID in Step 23c, determines the specific encoding of the signature input.

If applicable, the Steps 23c/23d may be repeated with a different ICC's private key and/or different terminal sector public keys.

Signature creation and verification algorithms, together with assigned object identifiers and encoding rules, can be found in ISO/IEC 20008-2, ISO/IEC 18370-2 and References [25] and [26].

7.6.3 Pseudonymous Signature of Credentials (PSC)

7.6.3.1 General Protocol description

The Pseudonymous Signature of Credentials (PSC) is another variant of the pseudonymous signature. An eID-Server may use this variant in combination with ERA (see 7.3.5 and Figure 8 Step 43) or EACv2 (see 7.3.3 and Figure 6 Step 24) in order to get a pseudonymous signature of on-card user attributes. This signature is a transferable proof of credential ownership to a third-party.

The eID-Server does not send an input message, but the concatenation of user attributes that are requested by the eID-Server are used by the ICC internally as input to the pseudonymous signature computation. PSC shall be available after EACv2 or ERA with dedicated access right (e.g. see Table 21 and Table 22 for authorization bits in cv-certificates).

7.6.3.2 Protocol sequence

See Figure 6 for EACv2 or Figure 8 for ERA.

7.6.3.3 Data types and relying entities

See Figure 6 for EACv2 or Figure 8 for ERA.

7.6.3.4 C-RP description

7.6.3.4.1 C-RP sequence

The C-RP sequence of EACv2 given in Figure 6 and the C-RP sequence of ERA given in Figure 8 is extended by two further commands denoted 24a-b and 43b-c, respectively (see Table 36).

Table 36 — C-RP description of PSC protocol

Step	Command – INS	P1-P2	Command data field, Le field	Response data field, SW1-SW2
EACv2 or ERA shall have been performed				
24a or 43b	MSE SET DST – ‘22’ SM activated	‘41 B6’	{‘80’ – L – OID of PSC protocol} – {‘E1’ – L – {‘73’ – L – Sequence of file IDs {‘04’ – 2byte file ID} – … {‘04’ – 2byte file ID}} {‘53’ – ‘02’ – ‘00FF’ specific attribute to include in PSC (optional)}} {‘84’ – L – reference of private key for PSC} Set protocol parameter, set data to be signed and set private key for PSC	Data field absent
24b or 43c	PERFORM SECURITY OPERATION –‘2B’ SM activated Function in P1: COMPUTE DIGITAL SIGNATURE	‘02 00’	{‘73’ – L – Discretionary data template {‘80’ – L – SP public key for domain-specific identifier (optional)}} Create pseudonymous signature of credentials and optionally domain-specific identifiers, see 7.6.3.4.2 for data to be signed	{‘73’ – L – {‘82’–L–PuK.ID.Sector1 (cond.) }– {‘83’–L–PuK.ID.Sector1 (cond.)} – {‘84’–L– (c||s1||s2)}} Pseudonyms signature and conditionally domain-specific identifiers s1 and s2 see (7.6.3.4.2)
eID-Server may perform any other protocol in accordance to effective access rights				

7.6.3.4.2 Step 24b or 43c — Compute Pseudonymous Signature of Credentials

The data to be signed by the ICC in PSC shall have the following structure:

- a discretionary data template, DO‘73’, containing a sequence of discretionary data objects, DO‘53’, with order as given by the file IDs or data object tags in DO‘E1’ in the command date field of MANAGE SECURITY ENVIRONMENT command in Step 24a or 43b;
- each discretionary data object, DO‘53’, contains the logical content of the corresponding data group or specific attribute, respectively.

Signature creation and verification algorithms together with assigned object identifiers and encoding rules can be found in References [12], [13], [25] and [26].

7.6.4 ABC-based signatures (ABC-Sig)

7.6.4.1 General protocol description

The mechanism of ABC-based signatures allows for obtaining a privacy-preserving signature over messages, credentials, user attribute or attribute statements. It is an alternative mechanism to Pseudonymous Signatures of Credentials (PSC). ABC-based signatures are similar to group signatures. However, the group is dynamically defined by the attribute statement used for generating the signature. Details about the cryptographic algorithms and a formal analysis are given in References [22] and [23].

The ABC-Sig mechanism requires generation of an ephemeral key pair used for key agreement. This key is signed using a pseudonymous signature scheme. Thus, the ABC-Sig mechanism allows for achieving unlinkability for the usage of an ICC-individual key pair to be used in Chip Authentication, whereas private and public keys are the same as for CA-ABC.

Based on the auxiliary data structure (see 7.3.3), the ICC fills in requested attributes and evaluates attribute statements analogous to how it would do when executing the COMPARE command for such statement. The operation is performed only if the effective authorization for the requesting terminal allows for doing so.

The data structure completed with the requested credentials and evaluation results for the attribute statements is used as input to the ABC-Sig mechanism.

An ABC-based signature shall always be preceded with CA-ABC. Because of that, it may often be sufficient to only sign with respect to the transaction-specific identifier or one of the sector-specific identifiers revealed in the preceding CA-ABC protocol instance.

An ABC-based signature can be verified by the eID-Server or any other party (e.g. an auditor) with the public key being common to all or a large subset of the ICCs in the system.

The ABC-based signature fulfills the unlinkability and pseudonymous signature as well as cloning protection property according to 6.4.

7.6.4.2 Protocol sequence

See Figure 7.

7.6.4.3 Data types and relying entities

See Table 16.

7.6.4.4 C-RP description

Table 37 gives the C-RP description of ABC-based signatures. It extends Table 17 by Step 24a-b.

Table 37 — C-RP description of ABC-based signatures

24a	MSE SET DST – '22' SM activated	'41 B6'	{'80' – L – OID of ABC-Sig protocol} Set protocol parameter for ABC-based signature computation, i.e. ABC-Sig-Msg or ABC-Sig-Cred	Data field absent
24b	PERFORM SECURITY OPERATION, COMPUTE DIGITAL SIGNATURE – '2A' SM activated	'AE AC'	{'73' – L – signature data {'80' – L – SP public key for domain-specific identifier (optional)} {'91' – L – current epoch (optional)} {'88' – L – message (optional)}} Create signature over message or credentials and attribute statements according to authorization and optionally compute domain-specific identifier(s) or transaction-specific identifier	{'73' – L – {'82'–L–PuK.ID.Sector1 or transaction identifier (cond.)}– {'83'–L–PuK.ID.Sector2 (cond.)} – {'84'–L–(c||s1||s2)}} Pseudonymous signature c and conditionally the domain-specific identifiers s1 and s2 or conditionally a transaction-specific identifier s1
	NOTE The usage of even INS code is deprecated. ISO/IEC 7816-8 specifies odd INS code (see also Table 36).			

7.6.4.5 Protocol-dependent descriptions

7.6.4.5.1 ABC-based signatures of messages

ABC-based signatures of messages (ABC-Sign-Msg) are used for signing messages with an ABC contained in the ICC. ABC-based signature of messages requires that the message be provided as a data object in the input to the corresponding PERFORM SECURITY OPERATION command. A hash of this message needs to be provided as part of the auxiliary data as part of the preceding Terminal Authentication protocol (see the EACv2 protocol description in 7.3.3 for details on this structure).

A further input is an epoch for supporting a revocation protocol for ABCs. Additional revocation methods may require the ICC to establish a secure channel with a third-party for updating relevant revocation information.

7.6.4.5.2 ABC-based signatures of credentials and attribute statements

ABC-based signatures of credentials and attribute statements (ABC-Sign-Cred) are used for signing credentials comprised on the ICC or attribute statements based on attributes on the ICC. The former corresponds to the analogous functionality defined by EACv2 using PSC and uses the same data structures for specifying the credentials. The latter uses an extension of those data structures and realizes the semantics of the COMPARE command and obtaining a signature on the result of this. Leaving the optional message field in the PERFORM SECURITY OPERATION command unspecified, an ABC-based signature over credentials and attribute statements is computed by the ICC.

The corresponding data structures are provided in the auxiliary data structure communicated and authenticated through CA-ABC of ABC protocol.

Annex A
(informative)

Use cases

A.1 Electronic identity

This generic use case deals with providing and authenticating certified attribute information of the card holder to relying entities in any offline or online interactions, while ensuring that the card holder is authorizing and giving consent to the release of the attribute information. The cardholder should be able to select which attribute information is to be revealed in a transaction and it should be possible that exactly this information, and no excessive information, gets revealed. That is, the data minimization principle should be enforced in this use case.

This is a generic use case and many other use cases are special variants thereof. This use case particularly captures any everyday online interactions that benefit from secure identity-related attribute information being provided to entities. The use case leverages the fact that the digital token comprises attributes, such as the name, address, or date of birth, that are frequently required in both everyday offline and online interactions.

A.2 Age-restricted remote or local services

Age verification, or the current lack thereof, is perceived by many governments as a major issue in the areas of online sales of age-restricted products and services, thus concerning online liquor stores, online wine stores, sales of fireworks, knives, or weapons online, online gambling or movies, online adult content access, and the like. The offline variant of this use case is access to age-restricted venues such as bars or purchasing age-restricted products (e.g. alcoholic beverages or cigarettes at offline cigarette vending machines).

A plethora of regulations related to the access of age-restricted material online or buying of age-restricted products is in place, depending on the country. Such regulations can be enforced easily offline, often based on driving licenses or other government-issued identity cards used as tokens to associate an age predicate with a party. However, as of today, there exists a major problem of enforcing age restrictions online and governments are struggling with putting proper enforcement procedures in place. As the regulations differ greatly by country, this has led to various age verification approaches to be used in practice of which few properly consider the protection of the privacy of the user.

For most of the offline use cases as well as online use cases, data protection in terms of data minimization and user control is crucial. Particularly, revealing the date of birth is, in general, not necessary from a perspective of age verification. Ideally, only a predicate over the date of birth attribute that is equivalent to a minimum or maximum age assertion (e.g. age greater than 18 years) and evidence for the correctness thereof, is revealed and suffices.

A.3 Domain-specific identification (pseudonymous authentication)

Using domain-specific identifiers under which a user acts instead of a single unique identifier is sufficient for most services. It is actually hard to consistently argue for the need of a single unique identifier a citizen acts under towards multiple parties in the public or private sector. As mentioned in another part of this document, certain countries prohibit the use of the same unique identifier with multiple public sector domains or the reuse of government-issued identifiers in the private sector. In those cases, it is mandatory to employ technologies that allow for the use of different pseudonyms.

As a special case, this use case comprises the creation of unique pseudonyms with certain (governmental) domains (e.g. tax services or health care). This special case helps prevent using a unique identifier across all government domains, which may be a legally required property in certain legislations.

A.4 Proof of place of residence

These use cases consider online or offline services that require the verification of (parts of) residence information like online (local) government votes, surveys or polls. Ideally, the necessary address information only is revealed in such interaction, not necessarily the full address (e.g. the transaction can reveal that the person is a local, lives within a certain street, quarter, city, region, province, or any other entity of locality of suitable granularity, or within a GPS coordinate range).

A.5 E-commerce

This use case captures a user authenticating certain attributes of their civil identity towards an online service provider (e.g. the name and address for shipping merchandise, instead of the merchant relying on unauthenticated information as is the case today).

Today's e-commerce retail industry struggles heavily with the problem of fraudulent transactions (e.g. in the context of identity theft and resulting credit card fraud). According to a recent report, a fraction of 2 % of international orders in online retail stores in the UK is fraudulent, resulting in substantial fraud-related cost for the system, eventually being covered by the user. Widely adopting the practice of authenticating crucial attributes of an e-commerce transaction, such as name and address, can help reduce fraud of such kind.

This use case is a special case of the Electronic Identity use case for the private sector.

A.6 E-government

This use case captures identification and authentication in a government interaction context, requiring a certain set of card holder attributes which typically identify the holder. Examples for e-government interactions include, but are not limited to the following:

— pension services;

— social services;

— health services;

— local services;

— tax declaration.

Privacy is met by revealing only necessary attributes (e.g. for properly identifying the user as required by regulation).

EXAMPLE The use of a unique identifier throughout all e-government interactions is often not necessary for privacy reasons. Instead, the creation of domain-specific identifiers (pseudonyms) with different parties fulfills privacy requirements. This depends on country regulations. Some countries require the use of domain-specific identifiers while other mandate the use of the same unique citizen identifier in all e-government interactions.

A.7 E-banking

The use case of electronic banking captures the secure login to an electronic banking account under the identity shared with the bank, without the need of an additional token being issued by the bank and by revealing only necessary attribute information for properly identifying the party as required by the business process.

EXAMPLE If a driving license is used as identification token, do not reveal further attributes such as driving privileges or the driver's height as contained in Data Group 1 (DG1) in today's International Driving Licenses (IDLs).

The use case can extend to opening a bank account using strong authentication based on government-issued documents, thereby saving the process overhead incurred through an offline identification agent for the initial identification.

A.8 Health data and emergency access

Health data are sometimes stored on or managed by health insurance cards but may be additionally stored on electronic identity cards or driving licenses for emergency access purposes. The use case of emergency access discusses the association of relevant medical and other information with a citizen or driver through an electronic identity card or electronic driving license and thus, making such information available to emergency services in emergency situations. The goal of having such information readily accessible in emergency situations is to help save lives in road and other accidents.

Health data are extremely sensitive PII and subject to additional legislation in multiple jurisdictions. Hence, the data is to be protected in a strong way against any unauthorized read attempts in any scenario. Only ambulance services, doctors, hospitals, and similar or related parties should be able to read the data. Due to the data being relevant in emergency situations with the card holder being possibly unable to give consent to the access, the access to such data is done without consent. For increased data protection, an accountability mechanism should be available enforcing that each access to such data is accountable, thus, helping prevent misuse. Particularly, the data should, due to its sensitivity, be protected against being read when using the card for online interactions in the context of a compromised host device. Examples of relevant information to be contained on the license include, but are not limited to, the following:

- allergies against medication in order to avoid wrong connotation;
- allergies in general;
- medical conditions;
- medication;
- relevant medical history;
- organ donation;
- emergency contact person.

A.9 Proof of derived token ownership

When two or more groups of users want to communicate to exchange data or deliver reciprocally digital services without disclosing their personal identification data, they can use authentication tokens (e.g. according to Random Hand-back Authentication (RHA) protocol[12].

In contrast to the use cases described in A.1 to A.8 that deploy protocols from Clause 7, this use case requires the deployment of further protocols for inter-device communication, for instance, RHA.

Authentication tokens may be delivered to groups of users to delimit a circle of trust. The two communicating entities owning such tokens and belonging to two different groups may generate a common secret and derive a secure channel out of it for any further secure exchange. As a general rule, a group accepts to communicate with another group only if the ICC group or subgroup identifier is recognized and if the secrets (session keys) generated on both side match. Each user belongs to a

[12] http://www.ego-project.eu/egotechnology/29-technology/137-new-challenges.