CEN

WORKSHOP

AGREEMENT

CWA 14170

May 2004

ICS 03.160; 35.040

Supersedes CWA 14170:2001

English version

Security requirements for signature creation applications

This CEN Workshop Agreement has been drafted and approved by a Workshop of representatives of interested parties, the constitution of which is indicated in the foreword of this Workshop Agreement.

The formal process followed by the Workshop in the development of this Workshop Agreement has been endorsed by the National Members of CEN but neither the National Members of CEN nor the CEN Management Centre can be held accountable for the technical content of this CEN Workshop Agreement or possible conflicts with standards or legislation.

This CEN Workshop Agreement can in no way be held as being an official standard developed by CEN and its Members.

This CEN Workshop Agreement is publicly available as a reference document from the CEN Members National Standard Bodies.

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

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

Management Centre: rue de Stassart, 36 B-1050 Brussels

Ref. No.:CWA 14170:2004 E

Contents

Foreword

Successful implementation of European Directive Dir. 1999/93/EC on a Community framework for electronic signatures [Dir. 1999/93/EC] requires standards for services, processes, systems and products related to electronic signatures as well as guidance for conformity assessment of such services, processes, systems and products.

In 1999 the European ICT Standards Board, with the support of the European Commission, undertook an initiative bringing together industry and public authorities, experts and other market players, to create the European Electronic Signature Standardisation Initiative (EESSI).

Within this framework the Comité Européen de Normalisation / Information Society Standardisation System (CEN/ISSS) and the European Telecommunications Standards Institute / Electronic Signatures and Infrastructures (ETSI/ESI) were entrusted with the execution of a work programme to develop generally recognised standards to support the implementation of [Dir. 1999/93/EC] and development of a European electronic signature infrastructure.

The CEN/ISSS Workshop on electronic signatures (WS/E-SIGN) resulted in a set of deliverables, CEN Workshop Agreements (CWA), which contributed towards those generally recognised standards. The present document is one such CWA.

The purpose of this CWA is to specify security requirements and recommendations for Signature Creation Applications. The CWA is intended for use by developers and evaluators of a Signature Creation Application and of its components.

This version of this CWA [Part] was published on May 2004.

A list of the individuals and organizations which supported the technical consensus represented by this CEN Workshop Agreement is available from the CEN Central Secretariat.

This document supersedes CWA 14170:2001.

Introduction

This document specifies security requirements and recommendations for Signature Creation Applications.

Sections 3 to 7 contain the definitions, modelling and technical introductions to the Signature Creation Application that are necessary to support the specification of security requirements. They do not contain requirements.

Sections 8 to 19 specify the security requirements for each functional component of a Signature Creation Application together with their rationale. Security requirements are always expressed in tabular form.

Annexes detail recommendations, and any supportive rationale.

Guidance on how to conduct a conformity assessment on applications and/or processes claiming conformance with this document is provided in CWA 14172-4 "EESSI Conformity Assessment Guidance: Part 4 - Signature creation applications and procedures for electronic signature verification ".

1. Scope

This document specifies security requirements and recommendations for Signature Creation Applications that generate advanced electronic signatures by means of a hardware signature-creation device. It is not required that they are based on a qualified certificate.

The signature-creation device (SCDev) addressed by this document must be implemented in a separate piece of physical hardware, with its own processing capabilities for PIN code verification and for performing cryptographic functions. Unless otherwise specified, this SCDev needs not be a secure-signature-creation device (SSCD), i.e. an SCDev that has been assessed as compliant with the requirements set in the Annex III of the EU Directive [Dir. 1999/93/EC].

Therefore advanced electronic signatures which are created by a signature creation application compliant with the requirements of this document fall under the provisions of Art 5.2 of the EU Directive [Dir. 1999/93/EC].

If, instead, an advanced electronic signature, that is produced with a Signature Creation Application conformant with the security requirements and recommendations specified in this document, is also based on a qualified certificate and is created by a secure-signature-creation device, that electronic signature is a Qualified Electronic Signature that complies with the provision of Art. 5.1 of the EU Directive [Dir. 1999/93/EC].

This document:

- provides a model of the Signature Creation Environment and a functional model of Signature Creation Applications;
- specifies overall requirements that apply across all of the functions identified in the functional model;
- specifies Security Requirements for each of the functions identified in the Signature Creation Application excluding the Signature Creation Device.

A Signature Creation Application is intended to deliver to the user or to some other application process in a form specified by the user, an Advanced, or where applicable a Qualified, Electronic Signature associated with a Signer's Document as a Signed Data Object.

This document is intended to be independent of particular technologies that might be employed in products. The following aspects are considered to be out of scope:

- generation and distribution of Signature Creation Data (keys etc.), and the selection and use of cryptographic algorithms;
- the legal interpretation of any form of signature (e.g. the implications of countersignatures, of multiple signatures and of signatures covering complex information structures containing other signatures).

This document specifies security requirements that are intended to be followed by implementers of SCAs.

2. References

The following documents contain provisions which, through reference in this text, constitute provisions of this standard.

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

For a specific reference, subsequent revisions do not apply

For a non-specific reference, the latest version applies.

[1]. ETSI TS 101 733 – Electronic Signature Formats;

[2]. PKCS#15: Cryptographic Token Information Standard;

[3]. EN 1332-4 Identification card Systems: Man-Machine Interface – Part Four “Coding of user requirements”;

[4]. EC Directive Dir. 1999/93/EC of the European Parliament and the council of 13 December 1999 on a Community framework for electronic signatures;

[5]. CWA 14169 - Secure Signature-Creation Devices, version 'EAL 4+';

[6]. CWA 14171 - Procedures for Electronic Signature Verification;

[7]. ETSI SR 002 176 – Electronic Signatures and Infrastructures (ESI) Algorithms and Parameters for Secure Electronic Signatures;

[8]. ETSI TS 101 903 – XML Advanced Electronic Signatures (XAdES).

3. Definitions

For the purposes of this document the following terms are used. Some of them are defined in the specifically referenced standards. Additional terms are also defined. :

Advanced Electronic Signature – an electronic signature which meets the following requirements:

a. it is uniquely linked to the signer;

b. it is capable of identifying the signer;

c. it is created using means that the signer can maintain under his sole control; and

d. it is linked to the data to which it relates in such a manner that any subsequent alteration of the data is detectable. [Dir. 1999/93/EC]

Certificate – an electronic attestation that links a signature verification data to a person, and confirms the identity of that person; [Dir. 1999/93/EC]

Certificate Identifier – an unambiguous identifier of a Certificate;

Certification-Service-Provider – an entity or a legal or natural person who issues certificates or other services related to electronic signatures; [Dir. 1999/93/EC]

Commitment Type – a signer-selected indication of the exact implication of an electronic signature;

Cryptographic Token – a personal security device capable of performing cryptographic operations. An SCDev is a type of cryptographic token;

Data Content Type - a signature attribute that expresses the encoding format of the SD;

Data To Be Signed – the complete electronic data to be signed (i.e. the signer's document and signature attributes);

Data to be Signed Formatted – the components of the DTBS which have been formatted and placed in the correct sequence for signing according to the requirements of the SDO type selected by the signer;

DTBS-representation – data sent by the Signature Creation Application to the Signature Creation Device for signing;

Electronic Signature – data in electronic form attached to, or logically associated with other electronic data and which serves as a method of authentication of that data; [Dir. 1999/93/EC]

Object Identifier – a sequence of numbers that uniquely and permanently references an object;

Personal Identification Number – a number that is used as Signer's authentication data;

Personal Identification Number Pad (PIN Pad) – a peripheral in the signature creation environment that is used by the signer to enter a personal identification number (PIN);

Pre-view – A general term used for the Presentation functions (SDP and SAV);

Qualified Certificate – a certificate which meets the requirements laid down in Annex I of the Directive [Dir. 1999/93/EC] and is provided by a certification-service-provider who fulfils the requirements laid down in Annex II of that Directive; [Dir. 1999/93/EC];

Qualified Electronic Signature – an advanced electronic signature which is based on a qualified certificate and which is created by a secure signature creation device (Note: definition based on art. 5.1 of the Directive [Dir. 1999/93/EC]);

Secure Signature Creation Device (SSCD) – a signature creation device that meets the requirements laid down in Annex III of the EU Directive [Dir. 1999/93/EC]; **Signatory / Signer** – a person who holds a signature creation device and acts either on his own behalf or on behalf of the natural or legal person he represents. Note: The term 'signer' is used throughout this document as a synonym; [Dir. 1999/93/EC]

Signature attributes – Additional information that is signed together with the SD;

Note: in ETSI TS 101 903 [8], "signed attributes" are called "signed properties".

Signature Creation Application – the application within the SCS that creates an electronic signature, excluding the SSCD/SCDev;

Signature Creation Data – unique data, such as codes or private cryptographic keys, which are used by the signatory to create an electronic signature [Dir. 1999/93/EC];

Signature Creation Device (SCDev) – configured software or hardware used to implement the signature-creation data; [Dir. 1999/93/EC]

Signature Creation Environment – the physical, geographical and computational environment of the signature creation system;

Signature Creation System – the overall system, consisting of the SCA and the SSCD/SCDev, that creates an electronic signature;

Signature Invocation – a non-trivial interaction between the signer and the SCA or SSCD/SCdev that is necessary to invoke the start of the signing process in the SCA/SSCD to generate the Signed Data Object. It is the 'Wilful Act' of the signer.

Signature Policy – set of rules for the creation and validation of an electronic signature, that defines the technical and procedural requirements for electronic signature creation and validation, in order to meet a particular business need, and under which the signature can be determined to be valid(TS 101 733 [1]);

Signature Suite – combination of a signature algorithm with its parameters, a key generation algorithm, a padding method, and a cryptographic hash function (ETSI SR 002 176 [7])

Signed Data Object – this contains the result of the SCA signature process consisting of the Signer's Document (SD) digital signature, possibly the SD or a hash of it and Signature Attributes. It is in a format specified by the signer selected Signed Data Object Type;

Signed Data Object Type – the type of the Signed Data Object (e.g. as specified in the ETSI Electronic Signature Format document TS 101 733 [1] and in TS 101 903 [8]), which specifies the resultant content and format of the SDO that is output from the SCA;

Signer's Authentication Data – data (e.g. PIN, password or biometric data) used to authenticate the signer to the SCDev and which is required to allow the use of the signature creation data held on the SCDev. The signer's authentication data may be referred to as 'Activation Data' in other documents;

Signer's / Signers' Document – the document for which one or more signers intend to create an Electronic Signature or for which an Electronic Signature was created;

Signer's Interface – a man/machine interface through which the signer controls the SCA and SCDev to create an Electronic Signature;

Trusted Path – A path between two entities or components within an SCA that provides integrity, authenticity and confidentiality;

Verifier – an entity that validates or verifies an electronic signature. This may be either a relying party or a third party, e.g. an arbitrator, interested in the validity of an electronic signature;

4. Abbreviations

CEN	Comité Européen de Normalisation (European Committee for Standardization)
CEN/ISSS	CEN Information Society Standardization System
CSP	Certification Service Provider
CSPC	CSP Interaction Component
CWA	CEN Working Agreement
DHC	Data Hashing Component
DTBS	Data to be Signed
DTBSR	Data To Be Signed Representation
DTBSF	Data To Be Signed Formatter
EC	European Commission
EESSI	European Electronic Signature Standardization Initiative I/O Input/Output
ETSI	European Telecommunications Standards Institute
ETSI SEC	ETSI Security Technical Committee
ETSI SEC ESI	ETSI SEC Electronic Signatures and Infrastructures
ETSI ESI	ETSI Electronic Signatures and Infrastructures Technical Committee
ISSS	Information Society Standardisation System
ODA	Open Document Architecture
PDA	Personal Digital Assistant
PIN	Personal Identification Number
QC	Qualified Certificate
SAC	Signer's Authentication Component
SAV	Signature Attribute Viewer
SCD	Signature Creation Data
SCE	Signature Creation Environment
SCA	Signature Creation Application
SCS	Signature Creation System
SD	Signer's / Signers' Document
SDC	SD Composer
SDO	Signed Data Object
SDOC	Signed Data Object Composer
SDP	SD Presenter
SHI	SCDev Holder Indicator
SIC	Signer's Interaction Component
SLC	Signature Logging Component
SSC	SCDev/SCA Communicator
SSCD	Secure Signature Creation Device
WS/E-SIGN	CEN/ISSS Electronic Signatures workshop

5. Signature Creation Functional Model

5.1 Signature Creation Objectives

The overall objective of a Signature Creation Application is to generate an Advanced or, where applicable, a Qualified Electronic Signature that covers the SD, the signer's Certificate or, where applicable, Qualified Certificate (or a reference to it), and, conditionally, the Data Content Type of the SD.

5.2 Model

A Signature Creation Environment (SCE) for the creation of Advanced Electronic Signatures includes a Signer interacting with a Signature Creation System. The Signature Creation System contains a Signature Creation Application (SCA), a Signature Creation Device (SCDev) or a Secure Signature Creation Device (SSCD) if a Qualified Electronic Signatures is to be created, with an associated Certificate or a Qualified Certificate if a Qualified Electronic Signatures is to be created, as shown in Figure 1.

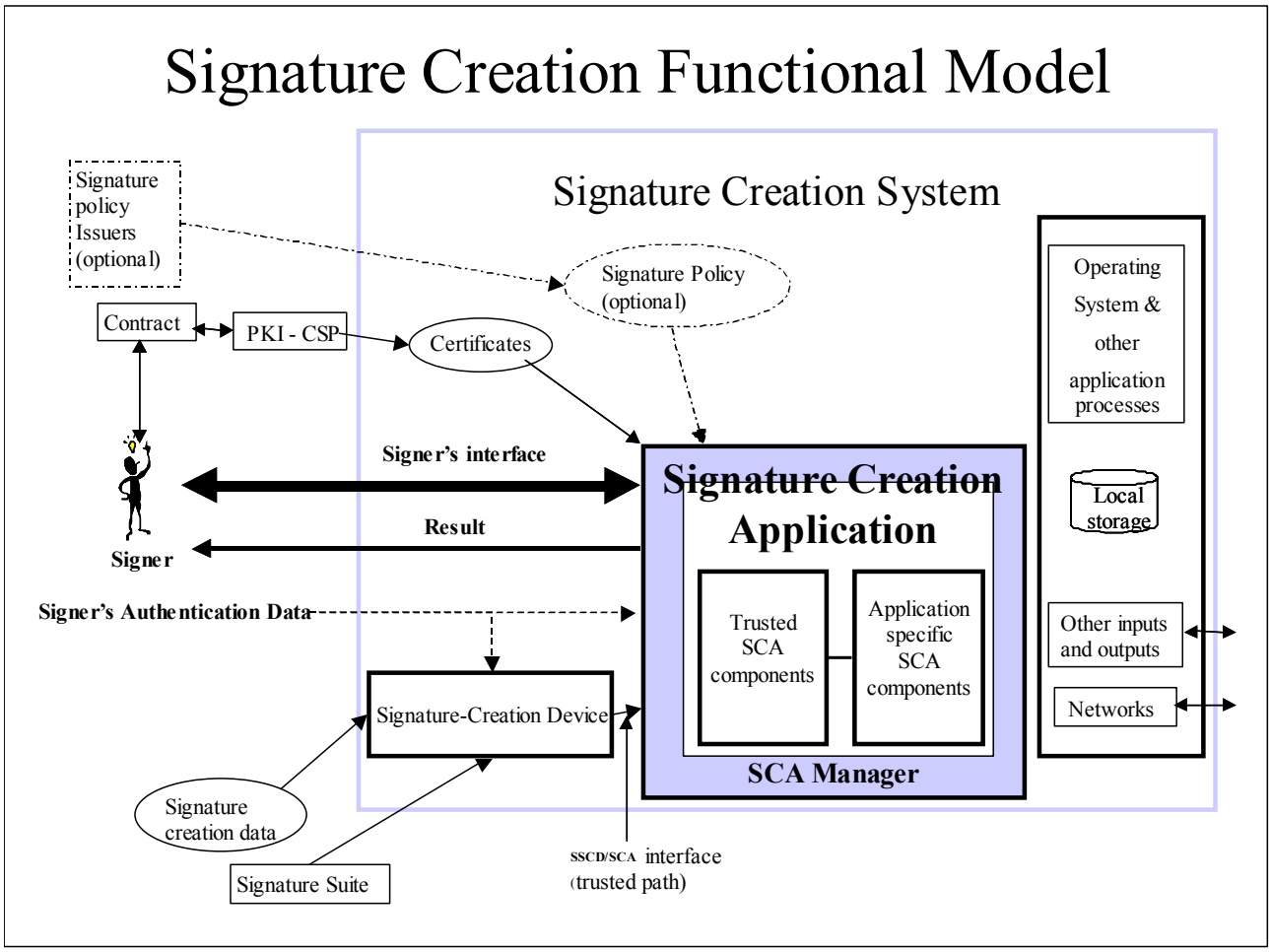

Figure 1 – The Signature Creation Functional Model

Figure 1 shows a functional model of a Signature Creation Application (SCA) as a part of a Signature Creation System within a Signature Creation Environment (SCE). It illustrates the signature creation functions and the information objects and interfaces that are relevant to its security. It does not distinguish between hardware or software implementations, and the model is not intended to specify the nature of any inputs/outputs or information transfer paths between the different functional components (which might take the form of direct I/O devices, hardwired connections or be distributed over communications links). Also, it makes no statement about the distribution of the functions over different platforms. These aspects can only become more concrete in the context of a particular set of technologies that apply to an SCS.

The purpose of the SCA and the SCDev is to take a SD and the related Signature Attributes, form them into Data To Be Signed (DTBS) and produce over them an Advanced, or where applicable a Qualified, Electronic Signature and to produce a Signed Data Object as a result.

The primary functions of the SCA are contained in a set of 'Trusted' and 'Application Specific' SCA components. These functions are elaborated further in section 5.3. In addition, the SCA will usually contain the following functions either to support the signature process or to support other functions that are not related to Electronic Signature Creation but which may have an impact on security requirements:

- An SCA Manager. This may perform a number of functions to support the signing process including the operation of the Signer's Interface, transfer of information from the Signer's Interface to the SCDev interface, interpretation of the Signature Suite and signature policy, obtaining the signature policy information and certificates, and management of local storage;
- The SCDev Interface. The SCDev is considered to be external to the SCA and will need to interact with the SCA to receive the Signer's Authentication Data and DTBS if there is no direct user interface between the SCDev and the signer, and return the digital Signature to the SCA;
- SCA local storage that may be used as a temporary location for data used during the signing process. This may also be considered as a target of security threats.

The SCA may contain other functions that are not related to signature generation e.g.:

- Data input/output ports and network connections that may be the target of security threats;
- Hardware/software processes that may also be the target of security threats;

The following information objects, which are all detailed further in section 6, are used within the SCE:

- A Signature Suite;
- Signature Attributes;
- Signature Creation Data;
- Signer’s Authentication Data;
- Signer's Certificates;
- The Signed Data Object;
- The SD;

The following interfaces and interactions are used to control the operation of the SCA:

- Selection of the document to be signed – to allow the signer to select the SD;
- Signature Attribute selection – to allow the signer to select the Certificate that is appropriate for the type of signature required and other signature attributes, ;
- Selection of the required Signed Data Object Type by the signer to specify the required form and content of the result (SDO) of the SCA;
- Signer’s Authentication Data input – to deliver the Signer's Authentication Data from the signer to the SCDev if the SCA is involved in this task;
- A secure presentation capability – to allow the signer to inspect the SD and Signature Attributes prior to invoking the signature process;
- An interface to CSPs – over which Certificates and, optionally, Certificate Revocations and Signature Policies may be obtained;
- Signature invocation – to allow the user to invoke the signing process (i.e. as a 'wilful act');
- SCDev interface – to enable the SCA and SCDev to communicate over a trusted path;
- An output – of the resultant Advanced, or where applicable a Qualified, Electronic Signature as specified by the Signed Data Object Type selected by the Signer.

The specifications for the Contract between the Signer and the CSP that provides the signer with a certificate. are beyond the scope of this document;

5.3 Signature Creation Applications

The primary parts of a Signature Creation Application for which this document specifies requirements are the set of trusted and application specific components that are shown in Figure 2.

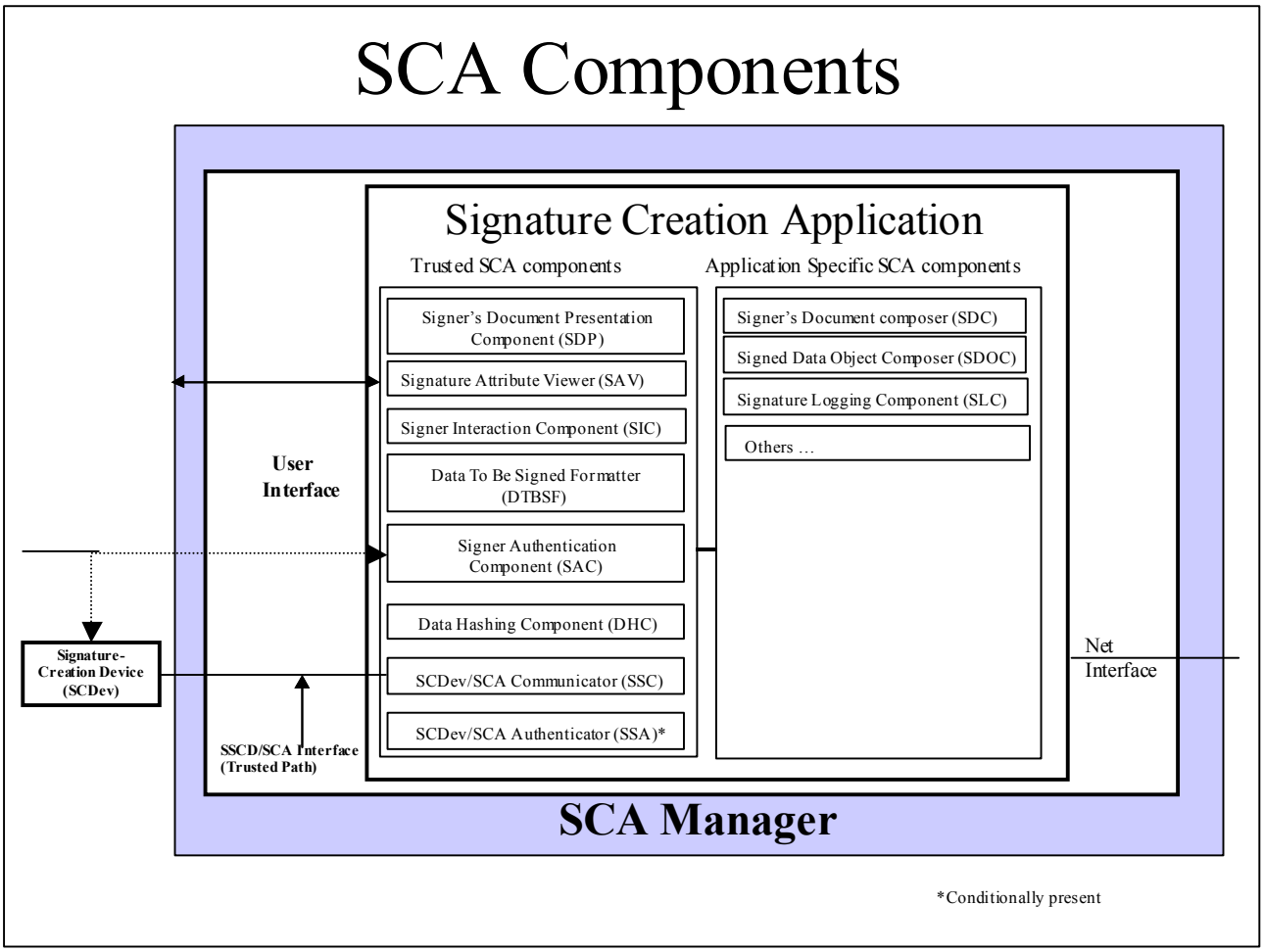

Figure 2 - SCA Components

A Signature Creation Application contains trusted and application specific components:

- The trusted components are all mandatory if not marked otherwise and are relevant for every SCA;

  Note: DHC and SAC are always considered to be present in order to encourage compatibility of the SCA with the widest possible population of SCDevs.

  The application specific components are application context dependent, i.e. their presence, construction and functionality is application specific.

The trusted SCA components are:

- SDP – SD Presentation Component used for presenting the SD that the signer selects by the Signer Interaction Component. The security requirements are specified in section 8;
- SAV - Signature Attributes Viewer used for viewing the Signature Attributes that the signer selects by the Signer Interaction Component and which will be signed together with the SD. The SAV shall include a capability to present the major components of the possibly application specific Signer's Certificate Content. The requirements are specified in section 9;
- DTBSF – Data To Be Signed Formatter which formats and sequences the SD or a hash of it together with the Signature Attributes and delivers the result to the Data Hashing Component. The requirements are specified in section 12;
- SIC - Signer Interaction Component through which the signer interacts with the SCA to control the signature creation process, and through which the SCA returns error and status messages to the

signer. This interface is used for all interactions between the Signer and the SCA, including input/selection of the SD and Signature Attributes except the Signer's Authentication Data. The requirements are specified in section 10;

- SAC - Signer's Authentication Component (e.g. a card terminal with PIN pad). This is used for presenting knowledge based Signer's Authentication Data and/or biometric features and preparation of the Signer's Authentication Data in such a way that they can be compared with Signer's Authentication Data held in the SCDev. The requirements are specified in section 11;
- DHC - Data Hashing Component for producing the DTBS Representation (which might be non-hashed, partially hashed or completely hashed as required by the SCDev). If the SCDev carries out all of the hash processing, then the task of this component is only to forward the DTBS Representation unchanged to the SCDev. The requirements are specified in section 13;
- SSC - SCDev/SCA Communicator which manages the interaction between SCA and SCDev. The requirements are specified in section 14;
- SSA - SCDev/SCA Authenticator which establishes a trusted path between SCDev and SCA. The presence of this component is conditional, i.e. it might only be present in SCAs that are under the control of public service providers and where the trusted path cannot be established by organisational means. The requirements are specified in section 15;

The application specific components may include the following:

- SDC - an SD Composer (e.g. a text editor) for creation, input or selection of the SD. The information that this acts on is managed through the SIC. The requirements are specified in section 16;
- SDOC - a Signed Data Object Composer that usually takes the DTBSF components and associates them with the bit string representing the digital signature as delivered by the SCDev, and outputs the result (i.e. the SDO) of the signing process in some standard format as specified by the SDO Type (e.g. as specified in the ETSI Electronic Signature Formats [1]). The requirements are specified in section 17;
- SLC - a Signature Logging Component that records some details of the most recent signatures created by the SCA. The requirements are specified in Annex Annex C;
- SHI - SCDev Holder Indicator that is used for displaying the SCDev holder's name. The requirements are specified in Annex Annex D.

In addition, section 18specifies requirements relating to communications between the SCS/SCA and the external environment.

Examples of devices which may support an SCA are:

- PCs;
- Laptops;
- Palmtops/PDAs;
- Mobile Phones.

The functionality of the SCA components and their related security requirements are outlined in sections 9 to 18.

5.4 Secure Signature Creation Devices

The SCDev performs those functions that hold the signer's signature creation data, verify the signer's authentication data and create the electronic signature using the signer's signature creation data. Examples of platforms on which SCDevs may be implemented are:

- Smart cards;
- USB Tokens;
- PCMCIA Tokens.

The interaction with an SCDev is described in section 14. The security requirements and a protection profile for SSCDs are provided in the CWA 14169 [5].

5.5 Signature Creation Application Instantiation

A Signature Creation Application Instantiation (SCAI) is a concrete implementation of the SCA components. One or more signature creation application instantiations (e.g. a secure e-mail application or a home banking application) may be present in the same physical unit, e.g. in a PC, and possibly share some SCA components.

5.6 Control and possession of Signature Creation Systems

Figure 3 illustrates two distinct cases of control and possession of an SCS in different types of SCE that result in different security measures needed to fulfil the security requirements.

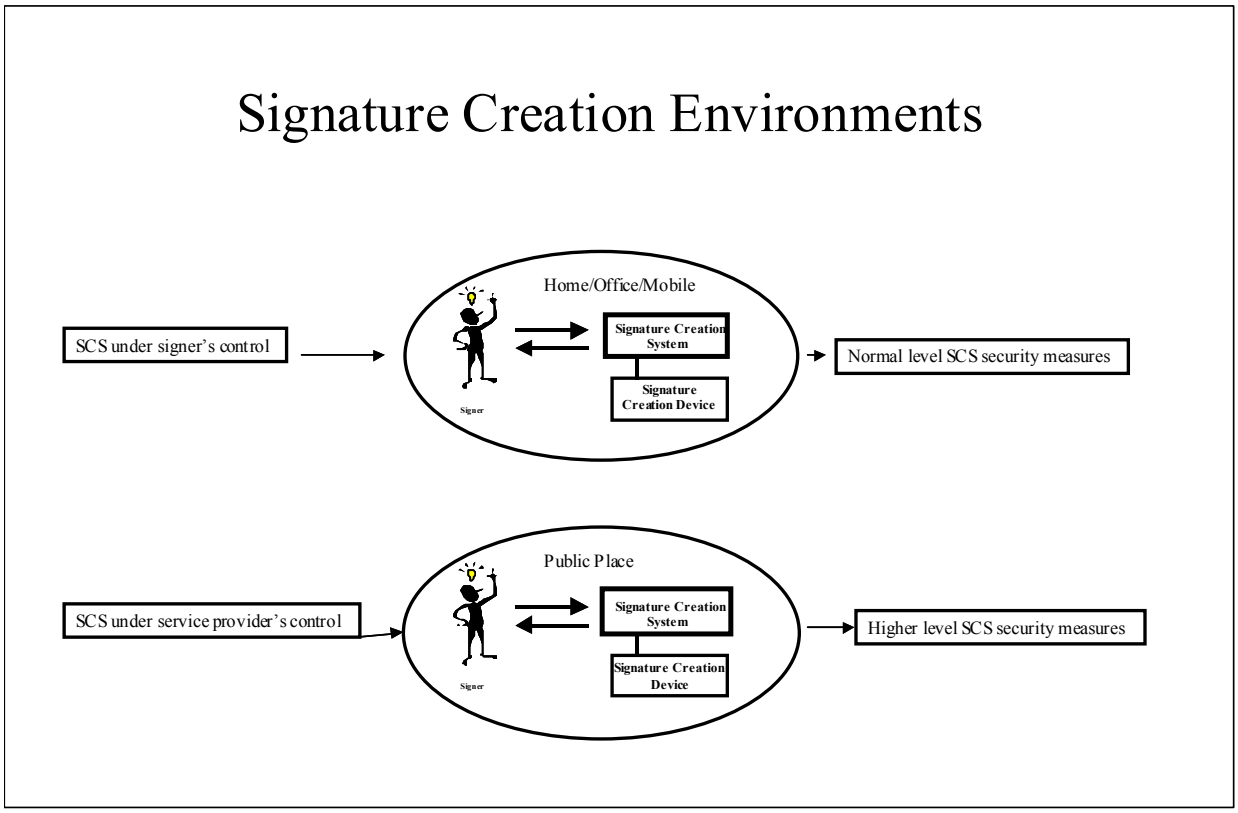

Figure 3 – Signature Creation Environments: SCS under Signer's Control and SCS under a Service Provider's Control

Figure 3 illustrates that the signing process can take place in two distinctly different types of physical environments where the SCS is controlled by different organisations.

A typical environment for the first case might be the home or the office, where the individual or the company has direct control of the SCS (e.g. an SCS implemented in a mobile phone). In this case, the security requirements may be met by organisational methods put in place or managed by the signer, and the technical means to ensure achievement of the security requirements may be more relaxed. For instance, in an extreme case, the Signer can use an isolated PC that is stored in a safe that can only be opened by the signer.

A typical environment for the second case is where an SCS is located in a public place such as a railway station, bank or any other SCS that is operated by a service provider that is not necessarily related to or under the control of the signer. Without further technical security measures, this type of environment can suffer a number of other types of attack - e.g. replacement with a fake SCS. The technical requirements of SCSs operated in such public environments will necessarily be more stringent.

These different environments have a different impact on the security requirements of the SCS since, although the overall security requirements remain the same for all SCEs as far as the signer is concerned, those security requirements need to be met in different ways.

6. Signed Data Object Information Model

Figure 4 outlines and relates a number of essential pieces of information for the generation of an Electronic Signature for an SD, and illustrates their relationship to the SCA components that generate them.

Annex A specifies some recommendations for the Information Objects. The following sub-clauses describe these information objects. No requirements are specified on how these objects are to be processed, provided that the security requirements defined in this documents are met.

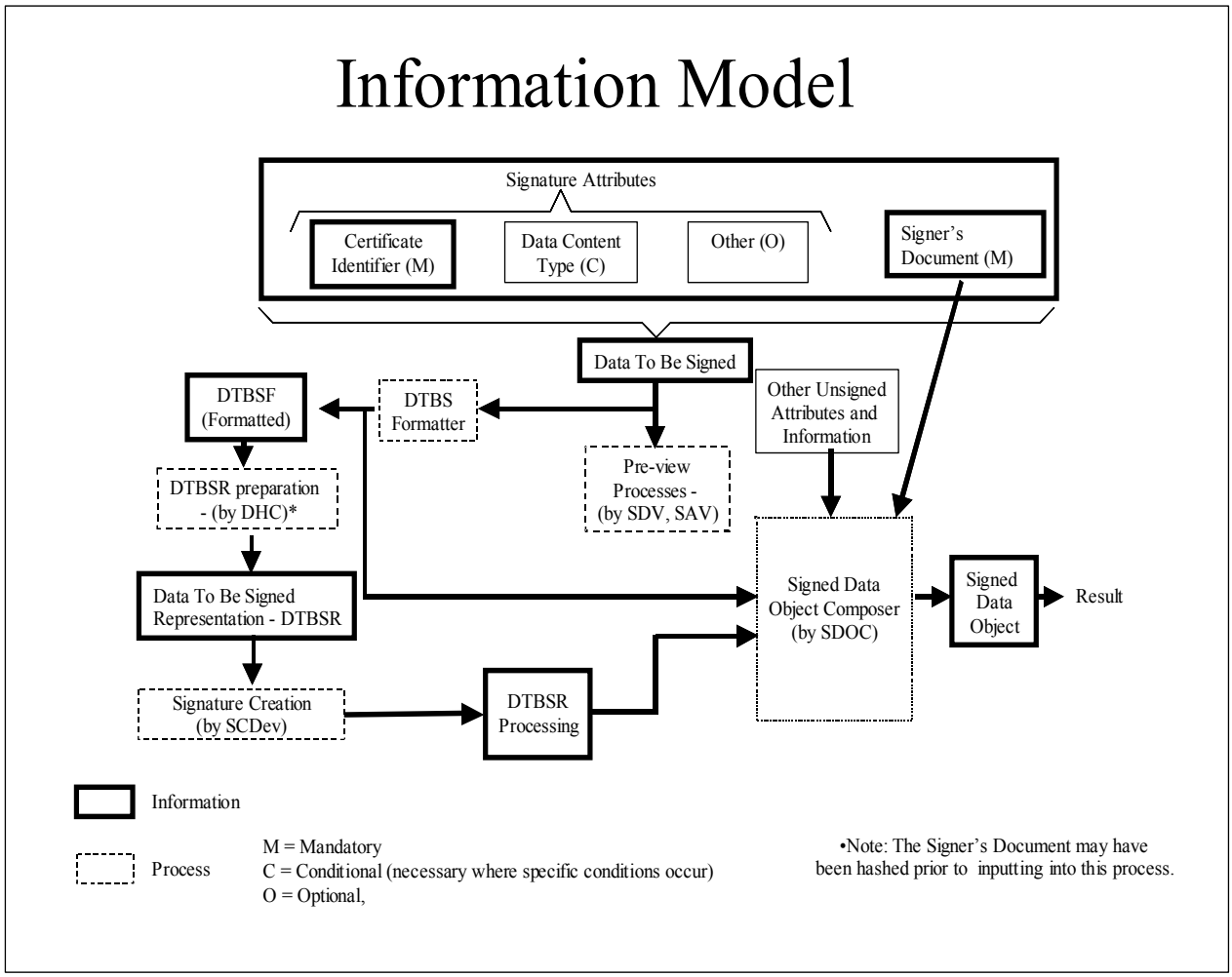

Figure 4 – Information Model of Advanced or, where applicable, Qualified Electronic Signature creation

6.1 Signer's Document (SD)

The Signer's Document (SD) is the document upon which the Advanced or, where applicable, Qualified Electronic Signature will be generated and to which it will be associated. The SD is selected or composed by the signer using the SDC component. In some cases, a hash of the SD may be presented to the signature processes instead of the complete SD.

The SD potentially has a number of important variants and components that impact the signing process and the status of the signature:

1. It may be in revisable format such as a word processor document or a message or file that can be edited, and where its presentation is dependent on the current configuration of the viewing device, and where the signer can potentially be presented a representation of the SD having an appearance different from that presented to the verifier;

2. It may be in an unambiguous, un-modifiable form (e.g. txt, Postscript, ODA final form …). These formats contain complete presentation rules that guarantee that the signer and verifier can be presented the SD in the same way if the same presentation rules are followed;
3. Hidden encoded information may be present (e.g. macros, hidden text, active or calculated components, viruses …). These may not be visible to the signer during the pre-view and verification processes, and the signer may not be aware of their presence. These represent potential ambiguities in the SD and are regarded as a security threat;
4. It may be in a form that is not normally presented to the signer or verifier directly, or it may be in a form that is inherently presented to the signer and verifier in different ways (whilst representing the same semantics). Examples of these formats are Electronic Data Interchange formats, Web Pages (HTML), XML, SGML, and computer files;
5. It may potentially contain embedded Signed Data Objects that have been created by persons or entities other than the Signer.

The format of the SD is described by the Data Content Type signature attribute. This attribute specifies exactly how it should be presented or interpreted by the verifier, and the type of application or presenter that a verifier should use in inspecting or using the SD.

6.2 Signature Attributes

Signature Attributes are pieces of information that support the electronic signature and which are covered by the signature together with the SD. The following lists some mandatory, conditional or optional signature attributes:

1. A mandatory Signer's Certificate identifier. It is the identifier of, or a reference to, the certificate holding the Signature Verification Data corresponding to the Signature Creation Data that the signer uses to create the electronic signature. Its presence is required because the signer might hold, either at the moment of the signature or in the future, a number of different certificates that relate to the same signature creation data, because it prevents substitution of the referenced certificate with another one with different semantics. If the signer holds different certificates related to different signature creation data it indicates the correct signature verification data to the verifier;
2. A Signature Policy reference can optionally be present if required by the signing context (e.g. in a specified trading agreement). This reference indicates to the verifier which is the correct signature policy to be used during the verification process. For instance, a signature policy might be used to clarify the precise role and commitments that the signer intends to assume with respect to the Signer's Document;
3. A conditionally present Data Content Type describes the format of the SD and specifies how it should be viewed and used by the verifier and as intended by the signer;
4. An optionally present Commitment Type. This is an indication by the signer of the precise meaning of the signature in the context of the signature policy selected by the signer (i.e. where electronic signatures can express different intentions of the signer). If a signature policy reference is present the Commitment Type might be selected from the range specified by the identified Signature Policy;
5. Further optional Signature Attributes may be specified e.g., the role assumed by the signer, the location of the signer where the signature was created, timestamps, archived certificates etc. These may be added to the DTBS to support the signature process and its interpretation and purpose, however, they are not necessarily relevant to the security and quality requirements of the creation of Advanced or, where applicable, Qualified Electronic Signatures. Further examples of this information and its uses are contained in the ETSI Electronic Signature Format documents [1] and [8].

The Signature attributes are either input or selected by the signer through the SIC component.

6.3 Data To Be Signed (DTBS)

The Data To Be Signed consists of the information objects that are to be covered by the Advanced or, where applicable, Qualified Electronic Signature. These are:

- the SD;

- the Signature Attributes selected by the signer that are to be signed together with the SD.

6.4 Data To Be Signed (Formatted) (DTBSF)

This contains the DTBS components that have been formatted and placed in the correct sequence for the signing process by the DTBSF component. It is this information object that is covered by the digital signature as the result of the signing processes and which is included in an Advanced or, where applicable, Qualified Electronic Signature and used in verifying the signature. The format of the SDO is determined by the SDO type that has been selected by the signer. One example of such an SDO Type is the ETSI Electronic Signature Format [1].

6.5 Data To Be Signed Representation (DTBSR)

This is the result of e.g. hashing the DTBSF according to a hash algorithm specified in the Signature Suite. It is produced by the DHC component. In order for produced hash to be highly representative of the DTBSF, the hashing function must be such that it must be computationally infeasible[1] to find *collisions*[2] for the possible signature life. It is to be noted that should the hash function become weak in the future, additional security measures shall be taken, such as applying Time Stamp Tokens.

6.6 Advanced Electronic Signature

An advanced electronic signature consists of an SDO together with either the signer's certificate holding the signer's appropriate Signature Verification Data or a reference to that certificate.

It is derived from the signature input (which consists of the DTBSR and possibly some padding) using the relevant signature algorithm and the Signature Creation Data associated with the chosen certificate.

6.7 Qualified Electronic Signature

A qualified electronic signature is an advanced electronic signature computed over the DTBSR by means of the signer's Signature Creation Data held in the signer's SSCD, which is associated with the signer's relevant Qualified Certificate.

6.8 Signed Data Object

This is the output of the SCA produced by the SDOC component and formatted according to the SDO Type. It will contain the digital signature issued with the signer's Signature Creation Data, and may additionally contain the following:

- The SD;
- The DTBSF;
- Additional supportive unsigned attributes and information such as timestamps, this is application dependent. TS 101 733 [1] and TS 101 903 [8] provide details on unsigned attributes, that are called unsigned properties in TS 101 903.

6.9 Signer's Authentication Data (not shown)

This is information supplied by the signer to the SCDev (possibly via the SCA) to authenticate the signer prior to commencement of the signing processes.

[1] This means that, given a specific hash value "h", it is computationally infeasible to find another DTBSF such that its digest equals "h".

[2] It is called a *collision* when the digests of two different inputs are equal.

7. Overall Security Requirements of the SCA

7.1 Introduction

The SCA shall use the SCDev to create an Advanced Electronic Signature covering the SD and Signature Attributes under the control of the signer. It is important that relevant Signature Attributes are included if it is to clarify the role and commitment that the signer intend in creating the signature.

Where in the following the term SCDev is used, it means that the requirement is valid both for SSCD and SCDev. Where the term SSCD is used, it means that the requirement is only valid when using an SSCD to create a Qualified Electronic Signature.

7.2 Trusted Path

7.2.1 Basic Trusted Path Requirement

The trusted path specifies requirements for the common infrastructure supporting the signature components of Figure 2 to protect the DTBS components the DTBSF and the DTBSR whilst they are within the SCA and during transfer to the SSCD. The Trusted Path protects against the following threats:

Title of Threat	Threat	Security Requirement
1. Accidental or malicious corruption of the DTBS components	The infrastructure processes used by the SCA accidentally or maliciously alter either the DTBS, DTBSF or DTBSR selected by the signer or other protocol data used between the SCA and SSCD	The SCA shall maintain integrity of: 1. the DTBS, DTBSF and DTBSR and all other information supplied by the signer 2. Protocol data flowing between the SCA and SSCD
2. Accidental or malicious breach of confidentiality of the Signer's Authentication Data or DTBS components or DTBSF	The SCA infrastructure reveals or copies the Signer's Authentication Data and/or DTBS components or DTBSF to unauthorised persons	The SCA shall maintain confidentiality of the DTBS components, the DTBSF and the Signer's Authentication Data

Table 1 - Security Requirements for a Trusted Path

7.2.2 Requirements for Public SCA

If the SCA is not always under the control of the signer (e.g. in a public signing terminal) the following requirement applies:

Title of Threat	Threat	Security Requirement
1. Disclosure or misuse of the Signer's Authentication Data or DTBS or DTBSF by a Public SCS operated by a service provider	A public SCS breaches confidentiality of the Signer's Authentication Data or DTBS or DTBSF.	1. The SCA shall securely erase all data related to a signature after having completed each signature processing 2. A public SCS shall not retain or copy these elements to any party not authorised by the signer.

Table 2 - Security Requirements of a Public SCS operated by a service provider

7.2.3 Referencing the correct SD and Signature Attributes

It is important that the signature is applied to the DTBS components that the signer selected and that there is no possibility for other DTBS components being accidentally or maliciously substituted after the signer has performed the pre-view and before creation of the DTBSR and initiation of the hashing function.

Title of Threat	Threat	Security Requirement
1. Substitution of one or more DTBS or DTBSF components	The SCA infrastructure accidentally or maliciously replaces DTBS or DTBSF components selected by the signer prior to completing the signature process	1. The SCA shall ensure that the DTBS presented to the signer are the same as those selected by the signer. 2. The SCA shall ensure that the DTBS components used to create the DTBSF and DTBSR are the same as those presented to the signer during the presentation processes and that they are identical to those selected by the signer.

Table 3- Security Requirements for referencing the SD and Signature Attributes

7.3 Requirements for Distributed Signature Creation Applications

The processes comprising an SCA may be distributed over different platforms. This implies that information may need to be passed over potentially un-trusted communications links or through potentially un-trusted system internal application program interfaces and software/hardware modules that render the Signer's Authentication Data, DTBS and DTBSF vulnerable to integrity, authenticity and confidentiality threats (e.g. through WEB browsers, mobile phones etc.). To counter these threats, the following requirements apply:

Title of Threat	Threat	Security Requirement
1. Breach of Integrity or Confidentiality of Signer's Authentication Data during transfer between SCA components	The Signer's Authentication Data is accidentally or maliciously corrupted, altered or its confidentiality is breached as it is transferred between SCA components	Any Signer's Authentication Data that is passed between distributed components within the SCA shall be transferred over a trusted path that provides integrity and confidentiality
2. Breach of Integrity or Confidentiality of the DTBS or DTBSF during transfer between SCA functions	The DTBS or DTBSF is accidentally or maliciously corrupted, altered or its confidentiality is breached as it is transferred between SCA components	The DTBS or DTBSF passed between distributed components within the SCA shall be transferred over a trusted path that provides integrity and confidentiality.

Table 4- Security Requirements for Distributed SCA Configurations

7.4 Requirements resulting from un-trusted processes and communications ports

Some systems and application processes, peripherals and communications channels that would normally operate in the same systems environment as the SCA (such as Operating Systems, Communications Systems/Services, Document/Information manipulation applications etc.), but which are not required during the signature creation process shall be considered to be un-trusted. The following threats and requirements arise:

Title of Threat	Threat	Security Requirement
1. Interference from un-trusted processes and communications ports of the SCA	Process and I/O ports of the SCA that do not play a role in the signature processing may corrupt or breach the confidentiality of Signers Authentication Data, DTBS or DTBSF or corrupt the processes of the SCA itself.	All un-trusted system and application processes, peripherals and communications channels that are not necessary for the operation of the SCA shall be prevented from interfering with the signing processes.

Table 5 - Requirements for protection against un-trusted SCA components

This trust might be achieved in different ways in different environments. For instance, there might be no need to specially secure an SCA or close down application processes and operating systems if the SCA is permanently isolated from communication networks and it is used solely in the home or office environment, or where it is used in other physically secure environments, so long as the SCA remains under the sole control of the signer. However, the responsibility for SCAs in public environments falls on the public service provider, and the security requirements may be met by either technical or organisational means. To give the signer's a degree of confidence, the SCA should display the name of the service provider and a statement that the SCA is suitable for the production of an Advanced or, where applicable, Qualified Electronic Signature.

7.5 Post signature verification of the Signed Data Object

Despite all of the security measures implemented in the SCA, some form of corruption or substitution of one or more DTBS components or DTBSF may take place. Therefore it is strongly recommended that signers are provided with the facilities to enable them to check that a verifiable electronic signature has been applied to the correct Signer's Data and Signature Attributes – i.e. as a check on the overall SCA and SCE's correct functionality.

In some circumstances, signers may choose to do this using a signature verification system as described in the related CWA 14171 [6]that specifies requirements for Signature Verification Systems, however, the present document does not mandate the availability or usage of such a signature verification system.

7.6 Requirements of the DTBS

The following table sets out the requirements of the presence of DTBS Components:

Title of Threat	Threat	Security Requirement
1. Generation of an inappropriate signature	That a signature is generated for a 'null' document.	The DTBS shall contain an SD.
2. Ambiguity of the signer's certificate implied by the signature.	That an attacker can associate the signature with another of the Signer's certificates that may imply semantics different from those intended by the signer.	The DTBS shall contain the Signer's Certificate selected by the signer and related to the Signature Creation Data that the SCDev uses to generate the electronic signature and intended by the signer.
3. Inappropriate presentation of the SD.	That a verifier perceives a SSD in a different form to that perceived by the signer because the verifier interprets the SD using a different Data Content Type to that used by the signer. This can happen in the context of any application or security policy that allows more than one SD Data Content Type.	The DTBS shall contain the Data Content Type of the SD if this is not indicated by other means.

Table 6 - Requirements of DTBS Components

Other Signature Attributes may be included in the DTBS dependent on the particular application.

8. SD Presentation Component (SDP)

8.1 Purpose

The SD Presentation Component is intended to provide reasonable trust that a document that is about to be signed is the one that the signer intends to sign, and that it has not been, nor will be, corrupted or modified. It does this by securely presenting to the signer the document about to be signed according to its Data Content Type.

A secure SD Presentation Component will be capable of presenting SDs of a limited number of Data Content Types. The SDP should issue a warning if the signer requests the SCA to sign a document of any Data Content Type that the SCA is not specifically designed to support. Data Content TypeHowever, the onus is entirely on the signer to select an appropriate Data Content Type for the SD, and to be able to determine whether the SCA complies with the requirements for the Data Content Type.

8.2 Background

Each SD should implicitly contain, or be associated with a Data Content Type that specifies the details of how the document is to be presented or used by the verifier. In addition to this, the present document distinguishes between two classes of SD:

1. Presentation Sensitive SDs: Where the semantics are derived from the document's presentation to the signer and verifier in person, and where its semantics are dependent on that accurate presentation. Example Data Content Types are those of word processed documents, text, Final Form Documents (from ODA);
2. Presentation Insensitive SDs: Where the semantics are completely or in part interpreted by some automatic process or software, and where presentation and rendition cannot be guaranteed to be the same for the signer and verifier. This may be due to inherent differences in the systems and software using the SD or due to the fact that neither Signer nor Verifier ever 'see' the document. Example Data Content Types are EDI Interchanges, computer files, Web Pages (HTML), SGML documents, XML and other mark-up language based documents.

Also, the source of the components making up the SD may be local (i.e. generated by software under the control of the signer), or remote (i.e. imported from some distant system and generated by persons other than the signer). In all cases, the following requirements should be met:

- The semantics of the SD, where necessary complemented with the Signature Attributes, shall be unambiguous. This means that the SD and Signature Attributes shall contain all relevant information to enable the verifying system to make the correct interpretation of their semantics;
- The SDP shall ensure that the syntax of the SD conforms to the Data Content Type specified for the document;
- The SD shall not contain hidden, encrypted or 'active' code such as macros that can modify or appear to modify the information prior to or during interpretation of the SD semantics by the verifier; alternatively the SDP shall have the possibility for the signer to disable the hidden code; the only permissible exception is when:

  1 the SDP warns the signer of the presence of any kind of hidden code;

  AND

  2 a viewer exists for the document format, such that any modification, occurred to the document presentation after the signature without affecting the signature cryptographic validity[3], triggers a warning to be displayed to the verifier; such viewer must be commonly known and publicly provided to any verifier by a trusted source without discrimination, and there must be a way for the verifier to ascertain its authenticity and integrity;

  - In some cases, the SDP may not be able to present the SD accurately to the signer (e.g. because of limitations of the presentation capabilities such as limitation of monochrome displays when used to represent colour). In these cases, the SDP shall warn the Signer.

[3] "cryptographic validity" means the ability to perform the successful verification of the signature by applying the specific algorithm.

Some burden will usually fall on the Signer to ensure the correct operation of the SDP and preparation of SD to achieve this aim.

In practice, these conditions might be more easily met if the signer chooses to limit the functionality of the SDC used to create the SD and also limit the SD's complexity. For instance, this might be achieved by using only ASCII text together with a declared character set and repertoire, or by using PDF documents where no hidden code (e.g.: in javascript) exists or it is disabled, and by taking legal advice on appropriate ways to represent the document's semantics.

The Presentation Sensitive and Presentation Insensitive cases are subject to different requirements.

8.3 Data Content Type Requirements

The syntax of information contained in the SD is specified by the Data Content Type attribute. The SCA must allow inclusion of this attribute to ensure that a verifier cannot mis-interpret the SD.

Title of Threat	Threat	Security Requirement
1. Mis-interpretation of the SD through lack of Data Content Type information.	The verifier can mis-interpret the SD by assuming that it is of a different Data Content Type to that intended by the signer.	The SDP shall allow inclusion of the SD Data Content Type either implicitly in the document or explicitly as an explicit Signature Attribute.
2. Syntax fail	The document does not conform to the syntax specified by the Data Content Type specified for the SD	The SDP shall warn the signer of this fact and allow the signer to abort from the signature process.
3. Signing a document with an inappropriate Data Content Type	The signer signs a document of a Data Content Type that the SCA does not support. In certain cases this may lead to ambiguity.	1. The manufacturer's literature should state what Data Content Types the SDP can correctly deal with. 2. The manufacturer's literature shall state the potential consequences if the signer declares a false Data Content Type; 3. The SDP shall warn the signer against creating a signature of any SD that indicates that it is of an unacceptable Data Content Type for the SDP.
4. Signing the wrong SD	The signature is applied to the wrong SD	The SDP shall ensure that the SD presented to the signer by the SDP is the same as the one that will be signed in the signature process, and is the same as that selected by the signer for signing.
5. Signing falsified components of the SD.	The signer un-knowingly signs other embedded Signed Data Objects with false or otherwise non-valid signatures created by others.	The presentation process shall inform the signer that other Signed Data Objects are embedded in the SD. (The SDP may optionally be linked to a signature verification system to allow verification of those signatures).
6. Accidental modification of the SD by the signer.	The signer accidentally alters the SD during the presentation process.	The SDP shall not allow the signer to change any part of the SD.
7. Inadequate SD presentation due to SDP	The SDP may not be able to present all parts of the SD because of	The SDP shall warn the Signer if it cannot accurately present all

Title of Threat	Threat	Security Requirement
limitations	limitations of hardware, software or configuration. This can cause the signer to be unaware of some aspects of the signers document that is being signed.	parts of the SD according to the Data Content Type.

Table 7 - Security Requirements of the SDP component with respect to the SD Content

8.4 SD Non-ambiguity Requirements

Where presentation of the SD is important (i.e. presentation is one of the means of conveying the semantics), the signer should ensure that the verifier is provided with enough information to be able to accurately present the SD to the verifier over a user interface. Otherwise the SD is ambiguous, and the verifier may be able to infer a meaning from the SD that is not intended by the signer. The following is a non-exhaustive explanatory list of aspects that could be specified (by means of the Data Content Type) to achieve non-ambiguity of a presentation-sensitive SD:

Fonts	Table layouts
Pagination	Paragraph formatting
Sound	Audio rendition
Image	Visual rendition

Table 8 - Examples of variable parameters of Presentation Sensitive SDs

In order to ensure that the SD is unambiguous, the following requirement applies:

Title of Threat	Threat	Security Requirement
1. Ambiguity of SD Presentation	The Signer signs a document that can be presented in different ways inferring different interpretations, and is therefore ambiguous, and the verifier can extract an unintended meaning from the SD.	Conditionally, the SCA shall allow the inclusion of a Data Content Type Attribute in the DTBS to ensure that presentation of the SD is unambiguous - i.e. to present it in exactly the same way as it was presented to the signer during the presentation process.

Table 9 - Security Requirements for non-ambiguity of Presentation Sensitive SDs

8.5 Requirements for Presentation Insensitive SDs

Where the semantics of the SD is not dependent on its presentation, sufficient information must be included in it or in a Signature Attribute to ensure that it is unambiguous.

The following requirements apply to Presentation Insensitive SDs:

Title of Threat	Threat	Security Requirement
1. Ambiguity of a non-presentable SD	An SD can be ambiguous due to insufficient information describing the structure and interpretation of its semantics	Conditionally, the SCA shall allow inclusion of a Data Content Type Attribute in the DTBS to ensure that only a single interpretation of the SD's semantics can be made.

Table 10 - Security Requirements of unambiguity of Presentation Insensitive SDs

8.6 Hidden Text and Active Code Requirements

SDs prepared by the SDC may potentially contain information that cannot be (or is not intended to be) presented to either the signer or verifier (e.g. macros, hidden text …). The type of information is dependent on the application that generates the SD. However, the inclusion of such information may be a source of confusion and ambiguity since, for instance, macros can automatically update information in the SD. The following requirement therefore applies:

Title of Threat	Threat	Security Requirement
1 SD alterations.	The SD may hold hidden code capable of modifying the signed document presentation without affecting its cryptographic validity, thus deceiving the verifier and/or the signer.	1. An SDP capable to make the signer sign only a static document format should be available. If it is not available: a. The SDP shall warn the signer of the presence of hidden code. b. A SD viewer must be available without discrimination from a trusted source, capable to warn of modifications occurred after the SD was signed.

Table 11 - Security Requirement for the Absence of hidden text, macros and active code in SDs

9. Signature Attribute Viewer (SAV)

The Signature Attribute Viewer is intended to ensure that the SD and the intent of the signature is unambiguous by presenting the Signature Attributes so that they can be inspected. It should be possible for the signer to examine all Signature Attributes, but in particular the signer must be able to check the content of the following:

1 The Signer's Certificate

2 The SD Data Content Type (if present);

3 The Signature Policy (if present);

4 The Commitment Type (if present);

Use of a revoked or expired certificate is a security threat since it can lead to creation of invalid signatures. An SCA should therefore check the validity period of the signer's certificates before signing. The SCA should also check the revocation status of the certificate. This can be achieved, for example, either by accessing the CSP's Certificate Revocation Lists, or by reference to an appropriate Online Certificate Status Provider service.

Title of Threat	Threat	Security Requirement
1 Signing a wrong Signature Attribute	The signature is applied to the wrong signature attributes.	1 The Signature Attribute presentation process shall allow the signer to view the Signature Attributes. 2 The SAV shall ensure that the Signature Attributes presented to the signer are the same as those that will be signed in the signature process, and are the same as those selected by the signer for signing.
2 Accidental or malicious alteration of the Signature Attributes by the SCA.	The intent of the SD is changed by accidentally or maliciously altering one or more of the Signature Attributes	1 The Signature Attributes shall be protected for integrity and authenticity. 2 The Signer shall be warned of the presence of any hidden text, macros or active code in the attribute[4].
3 Signing Signature Attributes that may automatically change before presentation to the verifier.	The code of the Signature Attributes contains active components that can alter their presentation or semantics.	1 The attribute viewer process shall warn the signer of the presence of any hidden or active components (word processor macros etc.) that are embedded in the Signature Attributes. 2 An attribute viewer must be available without discrimination from a trusted source, able to warn of modifications occurred to the attributes after the signature is created.

[4] Normally, such code should not be present; however if it is, then the Signer may not be aware of it.

Title of Threat	Threat	Security Requirement
4 Referencing an invalid certificate in a signature.	By making use of an expired or revoked certificate an invalid signature is created.	The SCA should verify the signing certificate validity period and revocation status by accessing the relevant certificate status information and, if it is found invalid, it shall prevent the signer from using the corresponding SCD.

Table 12 - Security Requirements of the Signature Attribute Viewer

In particular, the Signature Attribute Presenter must allow the Signer to inspect the Certificate that will be included in the signature. Certificates are protected by the signature of the issuing CSP. However, a Signer may have a number of different Certificates that are used for different tasks and in different roles (i.e. in a personal or official capacity and possibly from different PKIs). Also, different certificates imply different signature semantics (e.g. contractual signature vs. authentication only …). So, there are consequences of accidentally using the wrong Certificate, and the signer needs to ensure that the right certificate is used.

The following security requirement applies:

Title of Threat	Threat	Security Requirement
1 Use of the wrong Certificate	The SD may be associated with the wrong Signer's Certificate, causing the signer to enter into commitments that were not intended.	The SAV component shall allow the signer to inspect the major components of the certificate selected for inclusion in the DTBS.

Table 13 - Security Requirements for Certificate Presentation

10. Signer Interaction Component (SIC)

10.1 High level user interface principles

The following principles should be observed during design of the Signer's Interface:

- Suitable for the task - A dialogue system should be suitable for the task to the extent that it supports the signer for the most effective and efficient completion of the task. For example: A dialogue should only present the signer with those concepts and choices that are directly related to the signer's activities;
- Consistent - A dialogue system should be consistent in the way a system operates allowing the signers to improve their skills and predict the effects of their actions. For example: Control actions should have the same outcomes throughout the system; control sequences should have the same syntax; terms and labels should remain the same and display items should have a designated location;
- Conformity with user expectations - A dialogue should conform with the signer's expectations to the extent that it corresponds to the signer's task knowledge, education, experience and commonly accepted conventions;
- Controllable - A dialogue system should be controllable to the extent that the signer is able to control the interaction until the goal has been reached.
- Error tolerance - A dialogue system should be error tolerant. Despite evident errors in input, the intended result should be achievable with either no or minimal corrective action. Informative error messages should lead the signer forward;
- Individual adaptation - A dialogue system should be suitable for individual adaptation to the extent that it is constructed to allow for modification to the signer's individual needs and preferences. For example: Inter-sector/culture data elements on a smart card (EN 1332-4 [3]) may specify a signer's preference for language, colour, sound etc.;
- Provision of adequate status reports and error messages to the signer - This is important to ensure that the signer does not get confused and accidentally create an inappropriate signature e.g. by providing wrong DTBS components.

10.2 Signature Invocation

Prior to creating a signature, the SCA must determine that the signer really wants to create an Advanced or, where applicable, Qualified Electronic Signature, and that this cannot come about by accident. This can be achieved by the SCA (or SCDev) prompting the signer to commit a sequence of pre-specified non-trivial interactions over the SIC. In this document this is referred to as the 'Signature Invocation'.

A Signature Invocation is a signal from the signer to the SCA over the SIC component indicating that the signer is satisfied that the SCA is referencing the correct SD and the correct Signature Attributes as verified by the presentation processes, and that the signer wishes to create an Advanced or, where applicable, Qualified Electronic Signature covering them, i.e. to sign the document.

The SCA must ensure that each signature generated is the result of an explicit Signature Invocation. The following requirements apply:

Title of Threat	Threat	Security Requirement
1 Accidental invocation of the signature process	The user can accidentally cause invocation of the signature process	Prior to initiation of the signature process, the SIC shall request the signer to perform a non-trivial Signature Invocation interaction with the SCA that is unlikely to occur accidentally.

Table 14 - Security Requirements for obtaining the Signature Invocation

The number of signatures that may be created for each Signature Invocation will depend on the security policy defined by the user or user's organisation. For instance, a doctor signing many (150) prescriptions would not really need to perform 150 full signature invocations in a single session, but all 150 signatures must still be considered to have been generated as a single wilful act. However, an executive signing high value contracts may want to place a limit of one signature for each invocation.

10.3 Signature process inactivity timeout

It is necessary to prevent situations where the SCA and SCDev are in the state where the Signer's Authentication Data has been provided and the signer remains inactive for long periods of time. E.g. where the signer has been distracted from signature processing and another unauthorised person might possibly be able to complete the signature process on a modified or substituted SDs and Signature Attributes.

Title of Threat	Threat	Security Requirement
1 An unattended SDA permits unauthorised signatures generation	The signer provides the Signer's Authentication Data to the SCA/SCDev, and another (unauthorised) person takes control of the SCA and SCDev and can produce an unauthorised Electronic Signatures.	1 The SIC shall place a limit on the idle time the SCA neither interacts with the signer, nor it is processing.; 2 If this time limit elapses, then the signer shall authenticate again to the SCA.

Table 15 - Security Requirements for Inactivity Timeout

10.4 Signer Control Functions

Control functions allow the signer to control the signature process and the activities of the SCE and SCA. The following controls shall be provided:

- ability of the SCA to allow the signer to select the SD and Signature Attributes;
- ability of the SCA to allow the signer to complete the Signature Invocation to initiate the signature processes interaction between the SCDev and SCA;
- ability of the SCA to allow the signer to select the Signer's Certificate appropriate to the signature to be created;
- ability of the SCA to allow the signer to provide knowledge based or biometric based Signer's Authentication Data;
- Depending on the policy, ability of the SCA to allow the Signer to change the Signer's Authentication Data.

10.5 Retrieval of Signer's Characteristics

If the signer wants to use an SCA installed in a public place, then it is helpful to specify to the SCA signer's preferences such as:

- language preferences;
- preferences for displaying information as a result of disabilities, if these are relevant to the human interface of the Signature Creation Application.

EN 1332-4 [3] provides a coding scheme for achieving this on smart card terminals (ATMs, ticket machines, etc).

If the SCDev provides such data, the SCA should support it.

For SCA's under user control this feature may be less relevant because these systems will usually be customised to the signer's needs.

10.6 User Interface Aspects

In this Section basic requirements of the user interface are specified.

Additional recommendations are presented in Annex Annex B - Guidance to implement a User Interface .

The major requirements for the user interface are:

1 ease of use;

2 error tolerance, especially the capability to withstand operation errors or premature termination of the signing procedure by a signer without undermining the confidentiality or secrecy of the signer's personal data.

Title of Threat	Threat	Security Requirement
1. Signer's actions undermine the process safety.	A misguided signer can perform operations, or input data, in a wrong way, so that an attacker can capture confidential data or steal the signer's identity.	The SIC dialog must be as straightforward as the application can implement, to prevent the signer from creating security loopholes
2. Personal data revealed by signature process interruption.	When the user quits the signing station his/her personal data can still be available for an unauthorised person to see.	Screen displays shall be cleared of signer's personal data after a time limit sufficient to perform normal operations. The fields where the signer's data were displayed shall be overridden by other "neutral" data, to prevent latent images from being read.

Table 16 – Security requirements for the User Interface

11. Signer's Authentication Component (SAC)

11.1 General Aspects

According to its definition, an advanced electronic signature is uniquely linked to the signer(see Figure 5). A Qualified Electronic Signature is a particular case of an advanced electronic signature, then it implies that a Qualified Electronic Signature shall be linked to the signer.

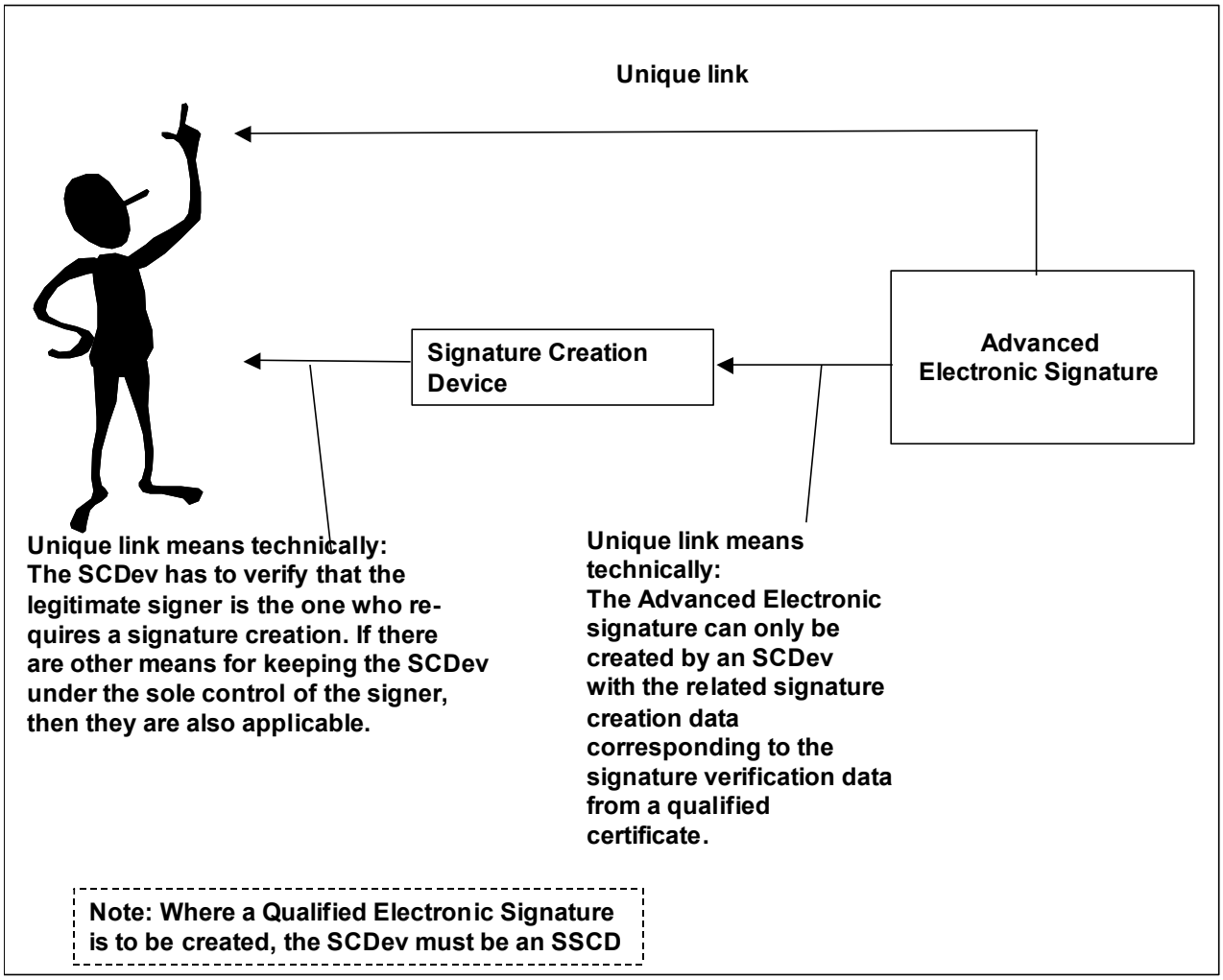

Figure 5 - Linkage of a Qualified Electronic signature to the Signer

The SCDev therefore performs an authentication procedure to verify that the legitimate SCDev holder is the one requesting creation of an electronic signature. Two types of signer authentication are possible:

- knowledge based signer authentication (i.e.: based on a PIN or password);
- biometric signer authentication.

After the signer has successfully presented the Signer's Authentication Data (e.g. a PIN, a password, a fingerprint), the "Security Status" of the SCDev is set to allow signing. Whether this security status is maintained, (i.e. the SCDev allows the creation of several signatures), or not (i.e. Signer Authentication is required for each signature created), depends on the signing application options.

11.2 Obtaining the Signer’s Authentication Data

Before creating an Advanced, Electronic Signature, the SCDev (and possibly the SCA) must make sure that the signer is the owner (or is authorised to use) the SCDev. It does this by obtaining the Signer’s Authentication Data from the signer. In some SCA/SCDev configurations, the Signer's Authentication Data

is passed from the signer through the SCA, and then transferred to the SCDev . If this is the case, then the requirements of Table 17 apply.

11.3 Knowledge based Signer Authentication

In a knowledge based authentication process, the Signer presents a secret to the SCA or SCDev. Examples of such secrets are

- a Personal Identification Number (PIN); or
- a Password (PW).

This secret is referred to as knowledge based Signer's Authentication Data. The SCDev compares this against a stored reference data copy of it held by the SCDev and produces a positive verification result if they match.

The secret must withstand guessing and brute force attack. This latter attack, even if a limit is set on the attempts, may be enacted in the long term if the attacker has regular access to the SCDev, and the unaware authorised user resets the attempts account by regularly authenticating to the device. Hence the need for a SCDev to set security measure requirements on the PIN and Password composition. Furthermore, the SCA must not prevent signers to modify their own secrets, where the SCDev allows it, should they realise they might have been compromised.

11.4 Biometric Signer Authentication

In biometric based authentication, the signer presents a biometric feature from which the biometric signer's authentication data is derived.

For some biometric systems, like those based on fingerprints, extraction of the template[5] is done inside the biometric terminal. The template is captured at registration time (enrolment), logically linked to the user, and kept either in a central data base or in a hardware token (the SCDev) carried by the user and presented at authentication time.

Attacks may forge the process result either at enrolment time (e.g.: by linking the user to someone else's template) or at authentication time (e.g.: by giving a fake "authentication successful" or "authentication unsuccessful" response).

In order to prevent these attacks it is highly recommended that extraction and storage of the template at enrolment time, and extraction of the template and its comparison with the reference one at authentication time, are done outside the SCA, where this is made possible by the used biometric method (e.g.: fingerprint based) and by the technology of SCDev , of its reader, and of the biometric sensor. Specifically, depending on the technology used, this may be implemented either entirely inside the hardware SCDev , or by means of a transaction between the SCDev and its reading device. What matters is that the biometric data are not passed through the SCA processing computer.

The strength of the biometric authentication mechanisms must be suitable for the protection of the signature creation data. This is reflected in the requirements of Table 17 and Table 18, which achieve the following:

- it is made difficult or practically impossible to make an impersonation attack with fakes of the biometric features;
- brute force attack are countered, e.g.: the reference data is protected by a retry counter;
- replay attacks are countered, e.g.: if biometric methods based on potentially publicly known data (face, ear shape, fingerprint) are used, then the Signer's Authentication Data is protected to ensure authenticity;
- it is made practically impossible at enrolment time to link a person to someone else's biometric template;
- it is made practically impossible at authentication time to alter the result of the signer's authentication data verification.

[5] The meaningful features of the biometric image and their correlations make up what is called a "template".

11.5 Provision of the wrong Signer's Authentication Data

The signer may occasionally provide the wrong Signer's Authentication Data (e.g. by mis-typing it). In many cases this is a genuine error. However, in other cases it might indicate that an attempt to discover the real Signer's Authentication Data by trial and error is in progress. Therefore, if the SCA is involved in handling the signer's Authentication Data, it shall indicate only the final result (i.e.: accepted or not accepted) of the signer's authentication process to the signer, and indicate whether the signer's authentication method has been blocked (i.e. that the SCDev will no longer perform a signer's authentication until the retry counter has been reset, if this is allowed by the security policy of the application provider). This leads to requirement 4 of Table 17.

11.6 Change of Signer's Authentication Data and Reset of the Retry Counter

Knowledge based Signer's Authentication Data, i.e. PIN or password, should be modifiable by the signer for two reasons:

1 an assigned PIN may not be easy to memorize by the signer (the signer is responsible for the quality of the new PIN or password, albeit the software that manages the PIN/password change can perform some basic controls to prevent trivial PIN/password from being set);

2 the Signer's Authentication Data may have lost their secrecy.

However, due to the security policy of the application provider or SCDev provider, the function for changing the related reference data in the SCDev may not be supported. Moreover, whether biometric Signer's Authentication Data can be changed, depends on the type of biometric method and the security policy of the application provider.

The number of comparisons with the Signer's Authentication Data should be limited, and be recorded with a retry counter to limit the number of erroneous presentations and prevent Signer's Authentication Data attacks. This also leads to requirement 4 in Table 17. The SCDev may also provide a means for resetting the retry counter to its initial value (e.g. by presenting a reset code, also referred as Personal Unblocking Key (PUK)).

11.7 Signer's Authentication Data User Interface Aspects

When presenting a PIN (e.g. at a PIN pad or keying in a password on a keyboard), then an appropriate feedback of a typed digit or character shall be provided (e.g. by display of a '*' or some other appropriate symbol or method). This leads to requirement 7 of Table 17.

The signer must be able to abort the presentation of the PIN or password.

If the PIN or the password is changed, then the SCA must require input of the PIN twice to ensure that no typing error occurred. This leads to requirement 8 of Table 17. In the case of biometric verification, exact user guidance is required. If a biometric method is used with a selectable biometric feature (e.g. right thumb or left thumb, right eye or left eye), then the required feature must either be indicated to the signer or, if there is a choice, selected by the signer.

11.8 Security Requirements for the SAC Component

The following security requirements are relevant for the SAC component.

NOTE 1: This document assumes that any environment requirements suitable to prevent attacks to the biometric device, such as submission of "fake" biometric elements (silicon fingers, usage of latent images, etc.) are in place.

Title of Threat	Threat	Security Requirement
1 Unauthorised use of the SCDev	An un-authorised parties gain access to the SCDev to forge signatures.	The signer must be provided with a means to input his/her authentication data to the SCDev or to the SCASCDev.
2 Disclosure of the Signer's Authentication Data by the	The Signer's Authentication Data confidentiality is compromised in the	If the Signer's Authentication Data are kept inside the SCA, it shall

Title of Threat	Threat	Security Requirement
SCA	SCA.	maintain the data confidentiality and securely erase it as soon as it is no longer needed (e.g.: they are substituted or the signer's enrolment is removed)
3 Accidental input of the wrong Signer's Authentication Data	The purported signer accidentally mis-types the Signer's Authentication Data	If the purported signer provides the wrong Signer's Authentication Data on a limited number of occasions, then an error response to the signer shall be generated and a retry permitted if the signer's authentication method has not been blocked by the SCDev. A corresponding message should be presented to the signer. No information on the type of mistake shall be provided to the user.
4 PIN/PW guessing	An attacker may obtain the PIN/PW either by guessing or by brute force attack.	The security measures are to be implemented by the SCDev, but the SAC must not prevent the management of PIN/PW by the signing device. Therefore it must: 1 be able to handle PIN/PW of the maximum length allowed for by the SCDev; 2 not prevent signers to modify their own PIN/PW at will.
5 Detection and misuse of knowledge based signer authentication data	If a PIN or a password can be detected on the path from the PIN pad or keyboard to the SSCD, then misuse of the SSCD is possible if the attacker gets access to it.	A trusted path for the transmission of a PIN/PW between the PIN pad or keyboard and the SSCD shall be provided through the SCA.
6 PIN/PW secrecy is compromised	If an attacker comes into possession of a SCDev a PIN or password, then misuse of the SCDev is possible.	A function for changing knowledge based signer authentication data shall be provided, unless it is forbidden for a dedicated SCA type according to the security policy of the SCA provider. (Note that the use of this function is only possible if the SCDev allows such a change)
7 Display of PIN or password	If the SCA displays the PIN or pass-word for signer feedback, then there is danger of eavesdropping.	A presented PIN or password shall not be displayed, but the feedback of a typed digit or character shall be provided to the signer by an appropriate symbol or method that does not reveal the PIN or Password.
8 Typing error by change of PIN/PW	If there is a typing error, the signer may not be able to input the correct new PIN/PW. Furthermore, this may lead to a weak PIN/PW.	The SCA shall require the presentation of a new PIN/PW twice and check whether both presentations are identical before delivering the new PIN/PW to the SCDev.

Table 17 - Security Requirements for knowledge based signer authentication data

Title of Threat	Threat	Security Requirement
1 Detection and replay of biometric signer authentication data	If biometric signer authentication data can be detected while it is presented to the SCDev , then misuse of the SCDev is possible if the attacker gains access to it.	A trusted path shall be provided for the transmission of biometric data between the biometric sensor unit and the SCDev
2 Misuse of public biometric signer authentication data	An attacker can get public biometric features such as face, and fingerprint, images and derive the Signer's Authentication Data from it in order to misuse the SCDev.	Biometric sensors must protect the user's biometric image from being used in replay attacks
3 Association to the wrong biometric authentication data	Malicious code can intercept the data of the person to be enrolled and link it to a biometric data belonging to a different person, to later on export this association that will be used by the impostor to impersonate the authentic user.	Biometric data association to the user should not occur inside the SCA computer.
4 False authentication.	An attacker can intercept the reply of the authentication process, in order to either give a false positive response (to authenticate an unauthorised person) or to give a negative response (to enact a Denial of Service attack).	Matching of biometric data should not occur inside the SCA computer.

Table 18 - Security Requirements for biometric signer authentication data

12. Data To Be Signed Formatter (DTBSF)

12.1 Functions of the DTBSF component

The DTBSF component takes the SD and the Signature Attributes and formats the Data To Be Signed (Formatted) - DTBSF. If the DTBS is to contain a hash value of the SD, and this does not already exist, then the DTBSF component initiates the hashing before production of the DTBSF. The reason for this is that in the selection of a document process, the signer might have directly selected the hash of a document.

12.2 Security Requirements for the DTBSF component

The following security requirements are relevant for the DTBSF component:

Title of Threat	Threat	Security Requirement
1 Wrong or incomplete DTBS production	Data Content Type 1 An attacker can prevent the SCA from applying all the signature components specific to the format chosen by the signer to achieve a given purpose (e.g.: long term validity). 2 An attacker can provide the SCA with forged signature components.	The SCA shall enforce controls to verify the validity, authenticity and completeness of all the components obtained in order to produce the correct DTBS format selected by the Signer.

Table 19 - Security Requirements for the DTBS component

13. Data Hashing Component (DHC)

13.1 Functions of the DHC Component

The Data Hashing Component takes the DTBSF and performs the following functions:

- production of the Data To Be Signed Representation (DTBSR);
- formatting of the DTBSR by conditionally adding possible padding for input to the SCDev and signature creation.

Work sharing between the DHC component of the SCA and the SCDev depends on the functionality of the SCDev .

13.2 Production of the DTBS Representation

The Figure 6 illustrates SCDevs with different functionality and their impact on security.

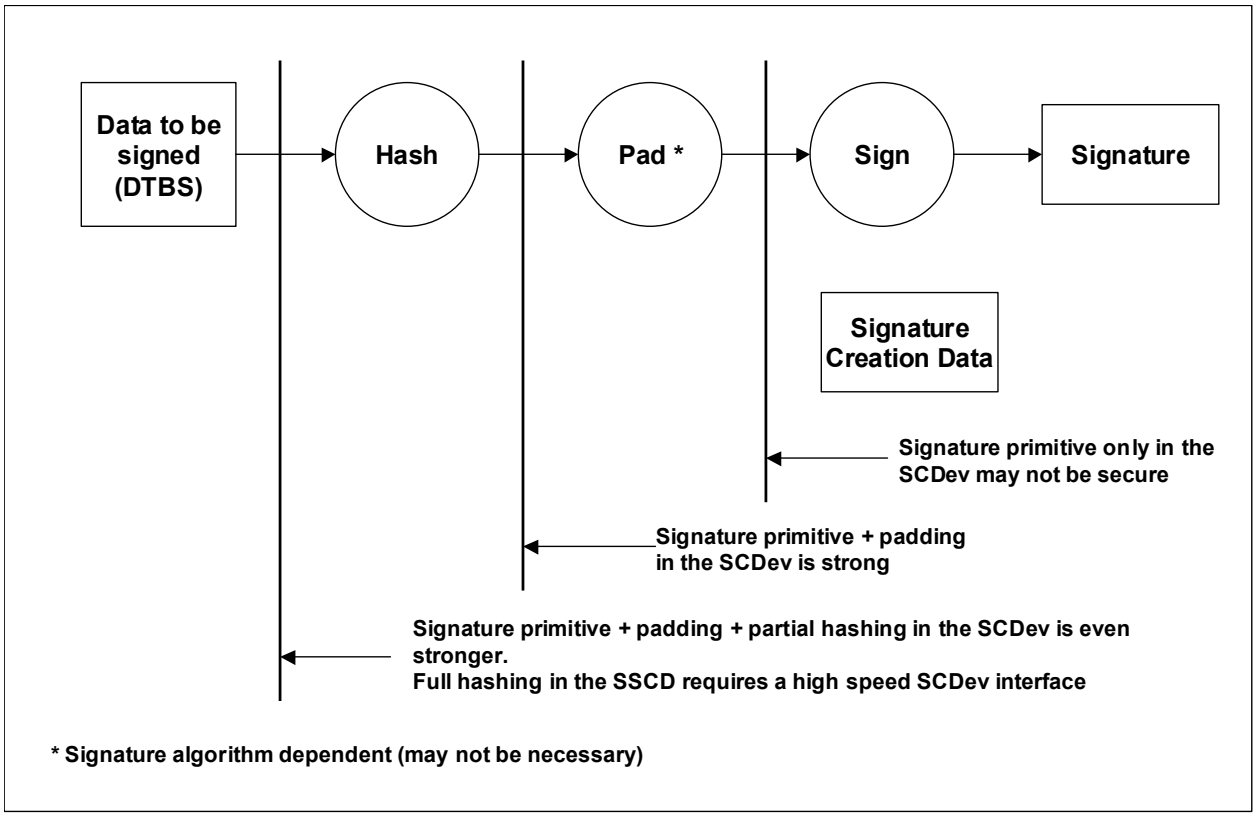

Figure 6 - Signature process and possible work sharing between SCA and SCDev

The first step after the user has invoked the signature creation process is hashing. Hashing is required since the message or document to be signed might have any length, but a signature algorithm can only process data that is less than or equal to the key length. The hash process is outside the visual and intellectual control of the user, i.e. the user is unable to calculate or recognise whether the hash value fits the DTBS or not. That means the user has to rely on this process step. The security requirement is therefore that the message delivered to the hash function is not modifiable (i.e. its integrity is protected) and that the hash function works properly.

The hash operation itself can be organised in different ways, as Figure 6 shows:

- hashing in the signature creation application;
- partial hashing in the SCA and completion of the hashing in the SCDev ;
- complete hashing in the SCDev .

Complete hashing in an SCDev of large documents is technically feasible if the SCDev has a high-speed interface like the USB (Universal Serial Bus Interface). SCDev

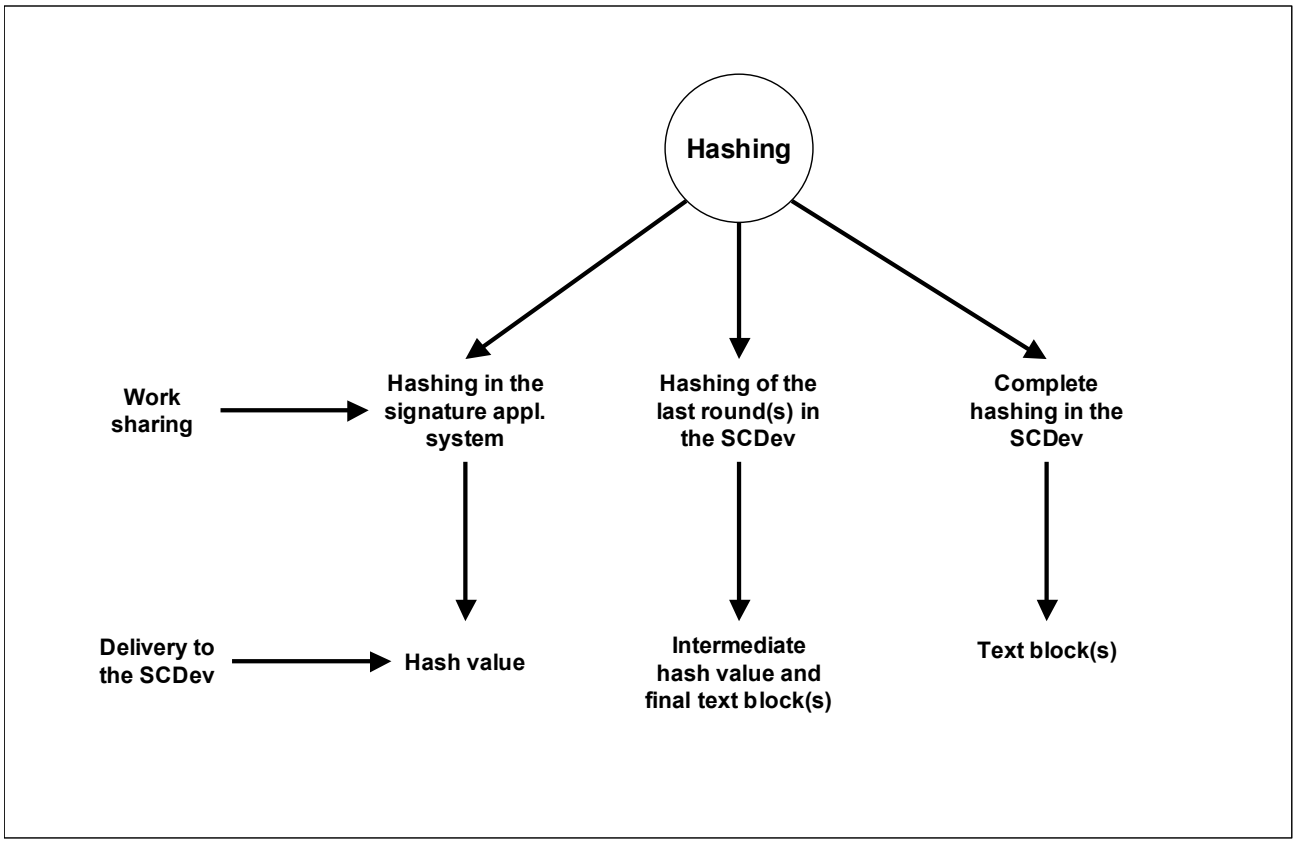

Figure 7 - Possible work sharing for hashing

Whether the respective SCDev expects that the hash value is delivered to the SCDev or whether the desired hash function is available on the SCDev should be indicated in the cryptographic token information.

13.3 Formatting of the electronic signature input

The second step in a signature process is formatting of the hash value. This is also referred to as padding.

Padding methods to be used in advanced electronic signatures are specified in [7].

To ensure proper interworking between a Signature Creation Application and an SCDev it is required that the Signature Creation Application knows what the SCDev expects as input for the electronic signature creation.

NOTE: An SCA may not support all types of input formats. If not, then the SCA is only capable of communicating properly with those SCDev s that support the input formats known to the SCA.

13.4 Security Requirements for the DHC Component

The following security requirements are relevant to the DHC component:

Title of Threat	Threat	Security Requirement
1 Weak hash algorithms	Weak hash algorithms may create collision.	The SCA shall ensure that only hash algorithms specified in ALGO document [7] are used
2 Weak electronic signature input formats	Use of weak signature input formats, which include padding, may cause the problem that the Signature Creation Data may be computable	The SCA shall ensure that only signature input formats specified in ALGO document [7] are used
3 Wrong or incomplete DTBSR production	If the DTBSR does not contain the mandatory and optional components required by the security policy and the signer, then it may lead to an incomplete and possibly ambiguously signed document.	The SCA shall ensure the production of the correct DTBSR for a signature.

Table 20 - Security Requirements for the DHC component

14. SCDev /SCA Communicator (SSC)

14.1 Interaction Sequences

The SSC component performs all of the necessary interactions between SCA and SCDev . Therefore from the viewpoint of security it is a very sensitive component, because any malfunction (e.g. due to attacks) may result in creation of a wrong signature. The interaction sequences are shown in Figure 8 and described in the subsequent clauses. Figure 8 also shows the difference in the interaction sequence performed:

- between an SCDev and an SCA under the signer's control;
- between an SCDev and an SCA under control of a Service Provider (which also indicates the difference between achieving the required level of confidence by organisational or by other means);

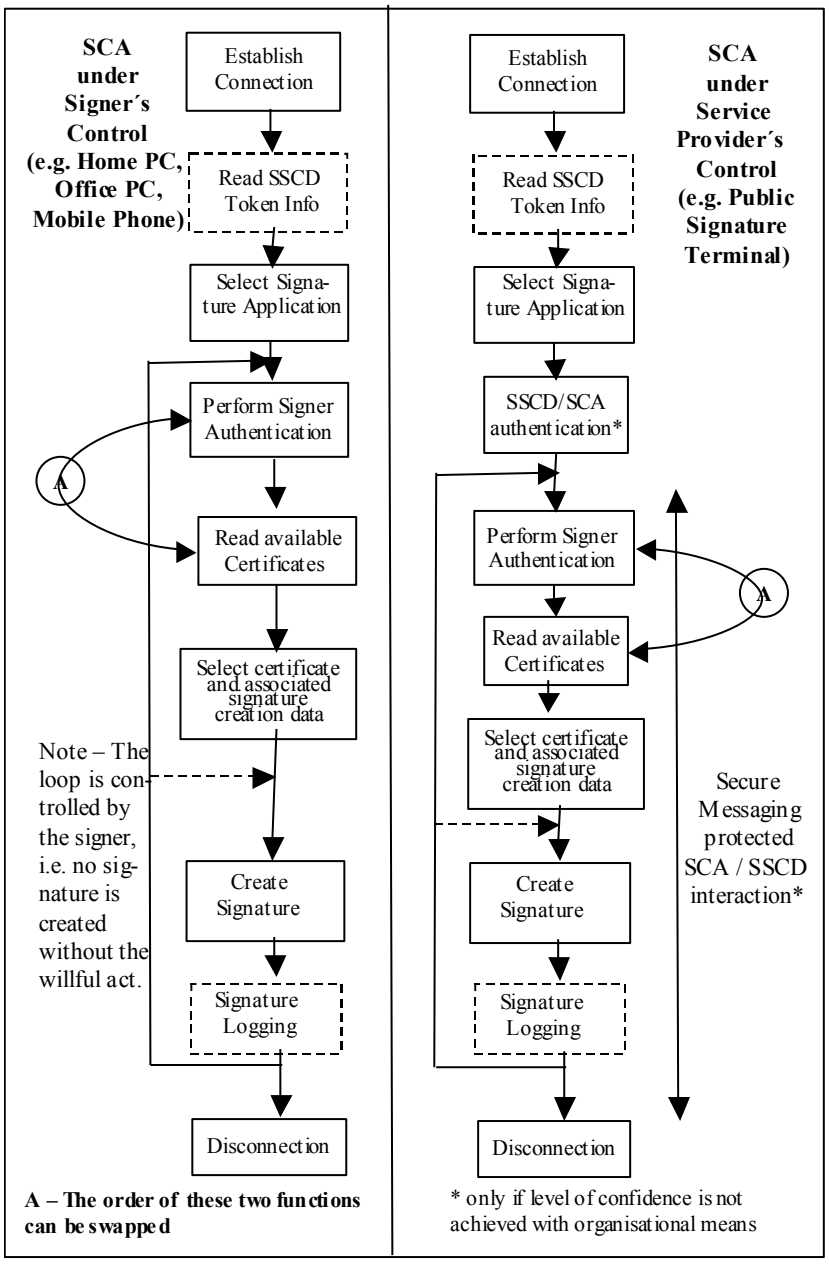

Figure 8 - Interaction sequences between SCA and SCDev

The dotted line with the arrow indicates that some SCDev s will require signer authentication prior to each signature creation and other SCDev s will be open for signature creation until disconnection or

closing of the signature application. The latter possibility allows the SCA to be more flexible, i.e. an SCA may be configurable in such a way, that it requires a Signer authentication:

- each time;
- after n-times;
- after a time period ‘x’ without creating a signature.

If an SCA allows configuration with respect to the authentication requirement, then the configuration function has to be protected, i.e. only the signer shall be able to modify it.

In the following clauses, the requirements for these interactions are outlined.

14.2 Establishing the Physical Communication

The SCA must have at least one physical interface suitable for communication with a SCDev.

For SCDevs permanently embedded within an SCA, the availability of an appropriate interface is required, however this need not be externally accessible.

For SCDevs where the connection has to be dynamically established e.g. by inserting in the SCDev , it should be ensured, that:

- sufficient power is provided and voltage is within the specified range to guarantee stable operation whenever the SCDev needs power;
- a clock with a frequency operating in the allowed range is available so that the SCDev can work properly and the transmission of bits can be synchronised;
- the bit convention of the SCDev is supported by the SCA (e.g. level "high" means a logical "1");
- the transmission protocol of the SCDev is supported by the SCA (e.g. character transmission protocol, block transmission protocol).

Example interfaces and related SCDev s are:

- smart cards which require a smart card interface where the card reader may be e.g. integrated in the PC-keyboard or the system unit, or attached as a separate card terminal to a suitable PC port (serial port, parallel port, USB port), or as a PCMCIA-card reader module for laptops;
- USB tokens which require a USB interface;
- PCMCIA tokens which require a PCMCIA interface;
- other cryptographic tokens which require e.g. a slot in a system unit with an appropriate bus interface.

The interface between an SCA and an SCDev may be e.g.:

- a contact link;
- a radio link;
- an infrared link;
- a combination of links.

Where radio or infrared links are used, the SCDev has to ensure that the signature creation function is either:

- not accessible; or
- the communication is organised in such a way that no additional risk occurs when using wireless interface mode.

14.3 Retrieval of SCDev Token Information

The EU Directive [4] for electronic signatures explicitly requires competition in the market for components and services relevant to electronic signatures to enhance user choice among various

kind of products and to allow migration due to the progress of technology. This has an impact with respect to the interface between a Signature Creation Application and a SCDev . Different types of SCDev s may vary e.g. in:

- the provision of signature algorithms (e.g. RSA, DSA, EC);
- the supported key length (e.g. for RSA keys 768 bit, 1024 bit, ...);
- the requirement for special formats of the signature input;
- the use of hash functions (none, SHA-1, RIPEMD160);
- the work sharing between SCA and SCDev with respect to hashing and signature input formatting;
- the method and type of user authentication;
- the provision of certificates;
- the types and sequences of commands for achieving the signature creation service from the SCDev

The presence of SCDev token information (e.g. the cryptographic token information in IC cards as defined in ISO/IEC 7816-15 – 'Cryptographic Token Information for IC Cards', which is compatible to the PKCS#15 [2] specification) is helpful for any SCA. However it is particularly useful for those SCAs that need to interact with different SCDev s, such as SCAs that are under control of a service provider. To achieve this, there is a need for the SCDev to provide information that enables the Signature Creation Application to deal with their SCDevparticular capabilities. This information states where data elements held by the SCDev are to be found and how they are to be used (e.g. the signature creation data, signer's authentication data, signer's certificates or a URL pointing to the certificates, if not stored in the SCDev).

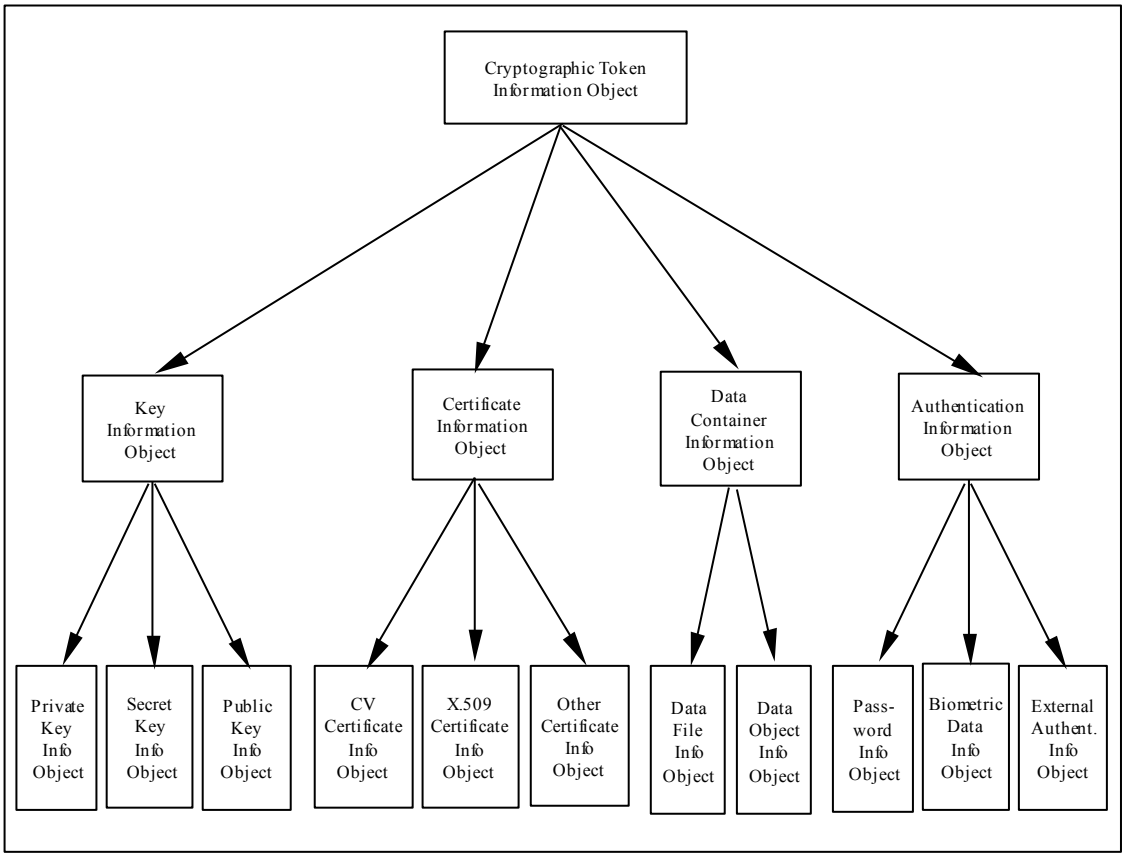

Figure 9 – Example: SCDev data elements to be described by the cryptographic token information objects

(figure according to the ISO/IEC 7816-15 on cryptographic token information)

14.4 Selection of the SCDev functionality on a multi-application platform

The SCDev functionality may be implemented on a platform (e.g. a smart card) which carries one or more SCDev functions (often referred as "SCDev Applications") and possibly other applications. Furthermore, the SCDev functions may be part of a larger application that has more functions than just the signature creation function (e.g. a home banking application). If such a multi-application platform is used as the carrier of one or more logical SCDev s, then the SCA must select one of them (e.g. by using the associated application identifier).

14.5 Retrieval of Certificates

An SCDev may carry several Certificates, e.g.:

- Certificates of the Signer used for different roles, different signature algorithms etc.;
- "Attribute Certificates" of the signer, if any;
- Certificates that may be useful for a verifier to build a certification path between the signer's certificate and a root key. E.g. Certificates produced by a root CA for the CA issuing the Certificates of the signer or by higher level AAs to the AAs issuing the signer's ACs, if applicable.

The SCDev should provide the following information (e.g. in the cryptographic token information) to the SCA:

- how to retrieve the certificates;
- the reference(s) of the related signature creation data;
- which certificates belong to which chain of certificates.

If an SCA under the signer's control already has the certificates stored, then they do not have to be retrieved again, i.e. an SCA may retrieve previously relevant certificates from an SCDev and store them so that a second retrieval is not needed (this saves time).

Depending on the security policy of the issuer of the SCDev, the retrieval of all or certain certificates may be always possible or restricted, i.e. retrieval of a certificate stored in the SCDev may be possible e.g. only after signer's authentication.

Dependent on the security policy of the issuer of the SCDev or the provider of the SCDev-application, the retrieval of all or certain certificates may be allowed or restricted, i.e. retrieval of a certificate stored in the SCDev may be allowed e.g. only after signer's authentication.

If the SCDev does not contain the certificate with the signature verification data (i.e. the public key of the signer) and possibly further certificates belonging to the signer's certificate chain, then at least an unambiguous reference to the signer's certificate in the form of a Uniform Resource Locator (URL) or another form of reference (specified e.g. in the cryptographic token information) should be retrievable from the SCDev.

14.6 Selection of Signature Creation Data

If an SCDev holds more than one instance of signature creation data, then the one appropriate for the signer's intentions has to be selected. Even if the SCDev has only a single signature creation datum, it may require that a reference to it is set. To enable the selection of the correct signature creation data, the SCDev Token Information has to contain information denoting the link between a certificate (possibly selected by the signer) and the signature creation data reference. If the SCDev also requires a reference to an algorithm, then this also has to be indicated in the SCDev token information.

14.7 Performing Signer Authentication

Where applicable, i.e.: where the SCDev has no SAD input device, the SSC component receives the Signer's Authentication Data from the SAC component over a trusted path and sends it with the appropriate SCDev command to the SCDev for comparison. The result shall be:

- verification successful; or
- verification failed; or
- verification method blocked due to e.g. too many consecutive faulty presentations of the Signer's Authentication Data.

The result is delivered back to the SAC component, which presents the result with an appropriate message to the signer.

14.8 Digital Signature Computation

The final step of the signature creation process is the computation of the digital signature (encryption of the DTBSR with the signer's SCD). In order to avoid usage restrictions, an SCDev should deliver a digital signature as a bit-string. The formatting of the relevant electronic signature and the results of the signing processes is context dependent and is a task of the SDOC component.

14.9 Signature Logging

If the SCA and the SCDev log completed signatures, then the relevant interactions between the SCA and the SCDev are performed after each signature creation has been completed. Further information is provided in clause 20.

14.10 Security requirements for the SSC Component

The following security requirements are relevant for the SSC component:

	Title of Threat	Threat	Security Requirement
1	Wrong signature through malfunction of the physical interface	A wrong behaviour at the physical interface, enforced or by system fault, may lead to a signature which fits a DTBS that is not intended to be signed by the signer	The SSC component shall support all items relevant to the physical interface in the specified range or with its specified characteristics to ensure proper operation of the types of SCDev that it claims to support.
2	Eavesdropping or interfering at a wireless interface between SCA and SCDev	Eavesdropping or interfering at the wireless interface may result in arbitrary security compromises	If a wireless or some other 'broadcast' link is used between SCA and SCDev , then the SSC component shall provide appropriate means for avoiding eavesdropping and interfering
3	Wrong selection of the signature creation data	The selection of an unwanted signature application or unwanted signature creation data in the SCDev (only relevant, if multiple SCDs are present in the SCDev) may cause serious problems for the signer since he may create a signature with a possibly unwanted certification path and inappropriate semantics	The SSC component shall ensure that the correct SCDev functionality is selected, if the platform, on which the SCDev functionality is implemented, requires a selection, and the related SCD according to the signer's choice of signature attributes is used.
4	Wrong signature creation due to SSC corruption	Any un-authorized modification of the SSC component may result in a wrong signature creation	The SSC shall be protected against any un-authorized modification.

Table 21 - Security Requirements for SSC component

15. SCD/SCA Authenticator (SSA)

15.1 SCA - SCDev Authentication for SCA under service provider's control

If the signature creation takes place at an SCA under control of a service provider (i.e. at a public SCA), then the signer needs to be able to determine whether to assume the same level of confidence as would be achieved if the SCA is under the signer's control. The confidence level for signature creation can be achieved by organisational means or by technical means.

Technical means might be:

- an SCDev authenticates the respective SCA and vice versa and
- the communication after authentication is protected by means of secure messaging, and
- the signer is able to recognise (e.g. by displaying a Signer specific display message) whether a secure interaction between the SCA and the SCDev can be assumed. The signer should be made aware that even this assumption might be insufficient in presence of malicious codes.

Note: If such an authentication procedure cannot be performed e.g. due to lack of verification keys, then this should be indicated to the signer. In any case the reliability of the authentication of an SCA by an SCDev might be affected by a malicious code that could intercept the dialogs between the SCA and the SCDev and between the SCA and the signer.

Since cryptosystems with symmetric keys may not be suitable for performing the authentication procedure due to key management problems and security risks, public key based authentication procedures should be applied as shown in Figure 10.

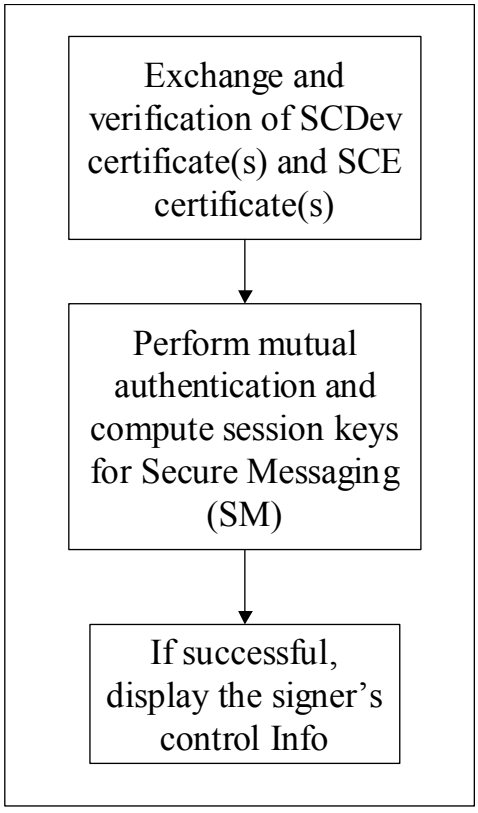

Figure 10 - Steps of authentication process between SCA and SCDev

15.2 Security Requirements for the SSA Component

The following security requirements are relevant for the conditional SSA component (i.e. it is only relevant for SCAs under a service provider's control and where the level of confidence is not achieved by organisational means):

	Title of Threat	Threat	Security Requirement
1	Compromise by a faked public SCA	A faked or modified publicly installed SCA may produce considerable harm to the signer	The SCA shall support entity authentication between SCA and SCDev , to provide a reliable indication to the signer of a successful authentication and protect the subsequent communication by secure messaging.

Table 22 - Security Requirement for SCAs under control of a service provider

16. SD Composer (SDC)

This function allows the signer to create or select the SDs that are going to be signed.

The SDC must not have the capability to include hidden code in the document version to be signed, alternatively it must have the capability to disable such code.

Note: the SDC is an application program able either to produce a document conformant with the previous requirements, or to parse an existing one and to clear it of possible hidden code.

16.1 Security Requirements for the SDC Component

The following requirements are to be complemented with requirements specified in section 8.6.

Title of Threat	Threat	Security Requirement
1. Hidden code inside the SD	A hidden code in the SD may change the signed text, without affecting the signature validity, in a way to deceive the signer, the verifier or both.	The SDC shall not allow insertion of hidden code in the SD. Alternatively it must have a feature to disable it.

Table 23 - Security Requirement for SDC

17. Signed Data Object Composer (SDOC)

This component takes the output of the SCDev (the digital signature) and associates it with the DTBSF according to the Standard Format determined by the SDO Type selected by the signer.

There are no security requirements here. However, the resulting Signed Data Object shall, as a minimum contain the Advanced Electronic Signature and may additionally contain the following components:

1 the SD;

2 the Signer's Certificate which corresponds to the signature creation data used to create the signature; ;

3 zero or more of the other Signature Attributes covered by the signature as determined by the SDO Type.

Also, it is essential that the chosen SDO Type specifies the sequencing of SD and Signature Attributes for which the Electronic Signature is valid, i.e. the same sequence as that used to create the DTBSF.

18. External Interface for Input/Output

18.1 Risks to the SCA

External input/output interface, (e.g. the connection to a public network like the Internet or an in-house network having links to public networks), are always a source of risks since e.g.:

- intruders may try to modify the SCA; and
- viruses that might have been imported with e-mail for instance, may corrupt data and software.

The signer or the service provider should therefore protect the SCA e.g. by:

- installing firewall mechanisms; and
- virus scan functions, or
- let all input be on intermediate devices with greater capabilities (e.g. better virus scanners)

18.2 Import of Certificates

If the SCDev does not contain all of the necessary certificates to create a signature, (i.e. it only contains the certificate identifiers), then the SCA should be able to retrieve these certificates.

The SCA should be capable of verifying the authenticity of retrieved certificates (please see also section 9).

18.3 Import of an SD and Signature Attributes

If the SD or part of it, or any Signature Attributes have been received over an input/output interface the SCA should ensure that no hidden parts play a role, or that substitution of any DTBS components cannot happen.

18.4 Download of SCA Components

In PCs and mobile equipment (e.g. with Internet browsers), it is common to use applets and plug-ins to enhance the functionality or to download e.g. a driver for an SCDev interface unit. If SCA components that need to be trusted are imported in such a way, then it must be verified that these components come from a trustworthy source, that their integrity is protected and that they are authentic.

18.5 Security Requirements for Input Control

The following security requirements are relevant for the input control if the security requirements cannot be achieved by other means:

Title of Threat	Threat	Security Requirement
1 Compromise of SCA components by malicious code	Imported malicious code may corrupt SCA components	Provisions shall be made to ensure that malicious code is prevented from corrupting SCA components or that SCA components that have been subject to a malicious code attack can be properly reorganised.
2 Compromise of SCA components by intruders	Intruders may corrupt SCA components	An SCA shall protect the integrity of its functional components and avoid the possibility that intruders can corrupt them.
3 Compromise if faked SCA components are installed	If SCA components are imported e.g. over Internet, then they may be a fake and cause the generation of in-appropriate signatures.	In the SCA provisions shall be made such that imported SCA components are only installed using a secure download.

Table 24 - Security Requirements for I/O control

Annex A (Informative) – General Recommendations

A.1 Operation of the Signature Creation Application

The signer, the SCA Components of the SCS and the SCDev co-operate to create an Advanced Electronic Signature of the DTBS in a series of steps. The following gives only a brief overview of those actions for a typical SCA and SCDev.

Some of the steps may be carried out in an order different to the sequence in which they are listed below.

1 The SCA is initialised into its operational mode. This might be achieved by the signer connecting the SCDev to the SCS or selecting and starting SCA software;

2 The SCA and the SCDev may then mutually authenticate each other, to assure the signer that the SCA can be trusted. The conditions for the requirements are specified in clause 17;

3 The signer then interacts with the SIC component to input, or select or compose the SD;

4 The SCA may then obtain the information from the signer regarding the Signature Attributes through the SIC component. The attributes actually selected will depend on the particular Signature Creation Application Instantiation, however, the Certificate Reference signature attribute is mandatory. The signer's certificate is associated with the intended role, name or pseudonym of the signer. This also allows the SCA and SCDev to select the correct signature creation data to generate the signature;

5 The SCA should then allow the signer to 'pre-view' the SD and the Signature Attributes using the SDP and SAV components to ensure that all of the information is present and correct. During this process, the SCA may also allow the signer to verify any existing electronic signatures that may be embedded in the SD (e.g. by using a Signature Verification System as defined in CWA 14171 [6]);

6 The signer may then provide the SCA with instructions on the type of signature to be created over the SIC component. This is the SDO Type. This is required if several different types of signature formats can be generated by the SCA (e.g. the different signature variants in the ETSI Electronic Signature Formats document [1]), or where different certificates from different PKIs can be used, and it is important for the SCA to know which type to generate and output;

7 The SCA must then interact with the signer over the SIC Component to obtain a Signature Invocation that instructs the SCA and SCDev to initiate the signing process;

8 The signer presents the Signer's Authentication Data either to the SCA or directly to the SCDev (e.g. PIN and/or biometric data) over the SAC Component. This is used to authenticate the signer to the SCDev, and to prevent persons other than the signer using the SCDev.

9 The DTBS Formatter component concatenates the SD (or a hash of it) with the Signature Attributes in the sequence determined by the required type of signature (SDO Type) to form the Data To Be Signed (Formatted) (DTBSF). The Signature Attributes are included in the DTBS to ensure that any attempts to substitute them with others, will cause a failure of the verification process;

10 The DHC Component takes the DTBSF and computes the DTBS Representation (DTBSR) using the hash process. In some cases all of the hash process is carried out in the SCA, in other cases some or all of the hash process may be carried out by the SCDev , dependent on the type of SCDev used. The DTBSR and optional padding is then passed to the SCDev over the SSC;

11 The SCDev then encrypts the DTBSR and possibly other data, e.g. padding, using the signer's signature creation data related to the certificate selected by the signer;

12 The output of the previous process is then passed back from the SCDev to the SCA over the SSC;

13 The SDOC component may then format the digital signature of the SD, the SD and the Signature Attributes, assembling them in an Advanced, or where applicable Qualified, Electronic Signature as per the SDO Type selected by the signer, and deliver the resulting SDO;

14 The signer may then be allowed to verify the electronic signature and 'post-view' the DTBS;

15 The formatted Signed Data Object may then be stored or sent from the SCA to recipient(s);

16 The signature activity may be logged by the SLC Component (i.e. certain facts about the signature are recorded either in the SCA, or the SCDev, or both);

17 The SCA may then be reset (i.e. for the case of an SCA operated by a service provider this shall involve securely erasing all data used during the signature process from the SCA processes and storage).

A.2 Requirement on the environment.

Closed Circuit Televisions shall not be located so that they can capture Signer's Authentication Data.

The SCS shall be located and designed such that it is not possible for others to observe / film Signer's Authentication Data.

A.3 Presentation insensitive SD

The following non-exhaustive explanatory list includes some types of document in this class and illustrates some aspects of the document types that could be specified to ensure non-ambiguity:

Type of information	Aspects
EDI transfers	Data Dictionary, encoding rules etc., and the application that should use the data
SGML, HTML, XML …	Document Type Definitions, Data Dictionary etc.
Files	File Type, File Format, File Structure, data element semantics etc., the applications that should interpret the data

Table 25 - Examples of variable parameters of Presentation Insensitive SDs

Annex B Guidance to implement a User Interface

B.1 Purpose

The purpose of this Annex is to provide guidance on the principles of constructing the user interface. This is in order to promote user confidence in the electronic signature creation process by making the interface as easy to use as possible, and by reducing the probabilities of human error without compromising security. It is also necessary to ensure that the interface is accessible for all users, including people with special needs; and by reducing the probability of human error.

This Annex therefore covers general user interface recommendations for the signer's interaction with the electronic signature creation process.

Determining user interface recommendations for information and communication technology systems, including electronic signature creation systems, is a result of the interaction of many different parameters, such as user's characteristics, tasks, applications, services used, technology and environments. This interaction determines a systems usability and acceptability. There is no single set of detailed recommendations that can cover all possible interactions.

The user interface consists of many different elements that should all fit together in order that the picture is complete. If an element is missing or substandard, the interface may not work. It is therefore important to see all the recommendations in relation to one another.

Note however that user interface requirements for hardware, e.g. correct ergonomic design of input / output devices such as keyboards, and local installation and environment (e.g. lighting conditions, physical accessibility to terminals, etc), are outside the scope of this document.

B.2 User interface consistency

Consistency in the user interface is acknowledged as one of the most important contributions to creating a user friendly system and at the same time reducing the possibility of introducing human errors.

There should be a consistent method for setting up, managing, using and receiving Electronic Signatures regardless of:

- which organisation provided the certificates;
- the different roles that may be played by the signer;
- what is being signed (e.g. voting form or bank agreement);
- which SCA is being used to create the signature.

B.3 Use of colour

Colour is one of the most powerful visual coding mediums that can be used to provide information. Its correct and consistent use is of considerable importance for user interface design. Employment of colour should consider its psychological associations, e.g. red = emergency/error, green = go/proceed.

Standard conventions for the use of colour should therefore be followed.

B.4 Feedback

Feedback is an important way of increasing the signer’s trust and confidence in a system and to avoid double entry of the same input data.

Feedback should therefore be consistently used and be provided in a timely manner to confirm that an action has occurred, that the action carried out by the signer is correct (or incorrect), and to confirm the status of the process (e.g. Please wait, system is checking your code/biometrics).

B.5 Security Breach detection

If the SCA detects a security breach, the user should be informed that the signature processing cannot go ahead under the present circumstances.

In the event of a security breach / conditions being inappropriate for signing, the system should provide an informative message to the signer and explain under what conditions the signer may recommence the signature creation process.

B.6 Invalid choice

Under some conditions, it may not be possible to select all choices that are usually available under normal conditions. The signer may try to select unavailable choices.

Therefore, any choices that are normally available, but which are not available under a particular set of conditions should be indicated.

B.7 Preservation of information presentation

Some systems support and provide full multimedia input/output and allow for adapting user interfaces to an individual's needs. It is important that the addition of a signature creation process does not adversely impact on such an existing system (e.g. prevent audio output).

The integration of electronic signature processing into to an existing system should not adversely affect its ability to provide information in a particular medium (e.g. text, image, audio, pictures and symbols, virtual 3D representations), nor to any human senses (inputs via text, language, voice, touch, gesture and other forms of motor control, biometric recognition).

B.8 Personalisation

Most computer operating systems allow users to set up different profiles, with different access rights so that the system is in some respect adapted to the individual. Public terminals (ATM's) can also be adapted to suit an individual's needs through coding of user requirements on a smart card.

The Signature Creation Process should be compatible with user profiling standards/techniques so that they could become an integral part of the user's profile.

B.9 Signer's Control when integrating with user profiling techniques

The potential advantages of integrating the SCA with user profiling techniques, must not mean that the signer looses control of the signature creation process.

The signer should therefore be able to override any user profiling technique so that they only make signatures that they explicitly intend to.

B.10 Configure /Edit Signature Creation process

Different terminals where Electronic Signatures can be created currently have completely different input/output functionality that in turn, will influence the dialogue the signer has with the system. A user profile (e.g. stored on a smart card as defined in EN 1332-4 [3]) could have preferred dialogues for different classes of terminal stored on the card. It could also have preferences for how the Signature Creation Processes could be handled on different terminals (e.g. if telephone terminal present then follow profile YY in languageXX).

The SCA should be compatible with user profiling techniques, and it should be possible to create and edit different SCP profiles in relation to different terminal types.

B.11 Distinguishing between certificates

In order to help the signer be able to clearly distinguish between certificates and remember what they represent, the signer should be able to assign to each certificate a label, symbol or icon that is meaningful to the signer.

B.12 Timing of operations

Peoples ability to read/listen, comprehend and react to messages varies considerably. Previous experience using a particular system also affects the time taken to react.

Sufficient time should be given to allow all users (especially the elderly and first-time users) to complete the signature operations. It is recommended that this be a user-defined time, where possible.

B.13 Security of terminals in public domain

Terminals placed in the public domain (e.g. shops, train stations) have inherent user interface/security problems such as:

- the ability of others/cameras to observe PIN values being entered;
- confidential information may remain on screens;
- confidential information may be left at the terminal (receipts);
- the terminal may be insecure (it may be a Fake), etc.

Some form of trusted information about the level of security of the SCA should be provided to the user. This could be by either organisational or technical means. The user interface of the device where confidential information (Signer's Authentication data) will be entered should be secured against visual observation by others/ cameras. Confidential/sensitive information should not remain on screen after user has finished. Voice /Sound output should not be used to provide sensitive/confidential information (e.g. codes).

B.14 User retention of secrets

Where secret codes are used (e.g. Signer's Authentication data) to control the SCA in making electronic signatures, as opposed to biometrics, they should be easily to recollect, i.e. short and/or meaningful. Where long codes are used, ways of helping the signer remember them should be considered, e.g. by «chunking» a string of digits, or by using a number series that is meaningful to the signer.

Codes should be easily retained by the signer. Ideally the signer should be able to choose the codes.

B.15 User instructions

The signer needs to know how to execute the signature creation process, what the consequences of his actions are, the importance of maintaining security (PIN codes), and who to contact if he suspects that security has been breached.

Unambiguous user instructions should cover both how to configure, install and use the system (ease of use) and security aspects. The development of instructions should be based on relevant ergonomic standards/guidelines. Guidance should be available in a readable and audible format.

B.16 Presentation of operational sequence

It is important that the user has a clear understanding of the various phases to be gone through and how to abort if necessary.

Information should be given to the signer clearly showing the different stages and activities to be followed in creating an electronic signature. This information may include numbering of sequence on screen, showing how far the user has progressed using flowcharts, pictograms, icons, symbols. It should be clear how/when the signer can quit/abort the process. The information should be communicable through visual, auditory and tactile information channels. For example, voice messages should be used to describe the transaction sequence and give audio confirmation of the keys selected.

B.17 Presentation of distinguishable parts

There are several component parts to creating an electronic signature that should not be confused, both with regard to ease of use and also with regard to understanding the consequences of processing these component parts.

The different component parts used to create an electronic signature should be clearly distinguishable from each other, e.g. by the use of at least two coding mediums such as use of colour, shape, labels, etc.

B.18 Guidance

Knowing that you will be safely guided through all the steps of a potentially complicated process is a prerequisite to building up confidence and trust in a system.

The next step in an operating sequence and associated functional areas should be clearly indicated.

B.19 Terminology

Technical terms will not be understood by most users. It is important that wherever possible everyday terms already in the public domain are presented to the end user, either on screen or in printed/spoken (e.g. cancel, clear and enter).

B.20 Error tolerance

Despite people making errors, either of omission or commission, the system should be robust enough to allow for this and not shut down or simply ‘crash’.

The electronic signature creation process should be tolerant of input errors (invalid entry) made by the user, e.g. if insufficient or excessive characters are entered into a fixed length string.

B.21 Informative error messages

There are many examples of unhelpful error messages to be found in ICT systems that are of no use to the user, and even blame the user for errors happening. Responses like “Error 213” are not very helpful.

In the case of invalid input, an informative error message should be given. The SCA should inform the signer that an error has occurred and how to remedy it. The message should not blame the user.

For example: If 10 numbers have been typed into an SCS but only 9 numbers are permitted, the SCS should respond with a message like "Please retype using a maximum 9 numbers". Information should be available in a readable and audible format.

B.22 Single handed operation of public SCAs

Some systems require users to hold several keys down simultaneously, an action requiring the co-ordinated use of both hands. For fully able-bodied persons this may not be a problem. However, for people who are not fully able-bodied, or who are holding something else (mobile phone, smart card / wallet in one hand) this task may not be so easy.

The process should not require two or more operations to be performed simultaneously i.e. operation should be possible using only a single hand.

B.23 Cancellation of operation

Knowing that you can safely quit an application at any time contributes to giving the user control over the system.

Signers should be able, at any time, to cancel the current operation and return to the main menu; or, to exit the system completely.

B.24 Undo operation

Given that people make mistakes, it is important that they can rectify these themselves without having to go through an extra sequence explaining that they have made a mistake.

There should be an “undo” function at appropriate steps in the signature process. It should be possible to undo only the last step, or the whole transaction.

B.25 Signer's Authentication Component (SAC)

B.25.1 Choice of signer authentication method

Different people have different preferences and abilities which in turn indicates a need to be able to input sensitive data in different ways. If a number of different signer authentication methods are supported by the SCA/SCDev combination, then the signer should be able to choose the authentication method (e.g. presentation of a biometric feature or a PIN).

Biometric methods may not be suitable, nor applicable to any Signer in the following cases:

- absence of the subject’s biometric feature;
- insufficient characteristics of the subject’s biometric feature;
- abnormal characteristics of the subject’s biometric feature.
- rejection due to personal reasons;
- cultural incompatibility.

Therefore, for an SCS under a service provider's control (a public SCS) that supports biometric facilities, a knowledge-based method should, for the time being, always be supported as a fallback alternative in addition to any biometric authentication methods.

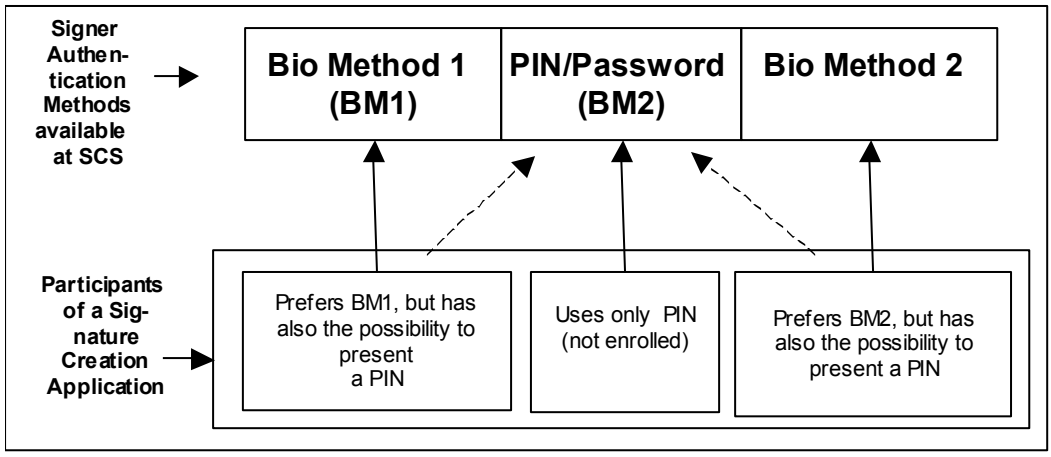

Figure 11 - SCA with different authentication methods (example)

In a signature creation environment where a 1-to-1 relationship between Signer and the SCA exists, an authentication method should be used that is appropriate to the Signer.

Where multiple signature creation data instances are present in the SCDev , a different signer’s authentication data may be required for each one. An indication of which signer’s authentication data belongs to which signature creation data can be provided by the cryptographic token information.

B.25.2 Biometric signer authentication

Examples of biometric features are:

- fingerprints;
- hand geometries;
- facial features;

- iris features;
- retina features;
- signature dynamics;
- voice patterns;
- key strokes;
- combinations of the above.

Some of the biometric methods are less relevant in a signature creation environment, however an evaluation of the suitability of the various methods is out of scope of this document.

Annex C Signature Logging Component (SLC)

For the signer, it is useful to get support from the SCDev and/or SCA with respect to signature logging, i.e. for each created signature, a logging record may be stored in the SCDev as Figure 11 shows. Since there is insufficient storage for an indefinite number of log records on a SCDev , a file with a cyclic structure is most suitable, so that at least a number of the last signatures can be logged. The information to be logged depends on the capability of the related SCDev and the concept of the application provider.

Examples of Logging information that might be considered are:

- Count of signatures generated;
- Date and time of signature creation;
- Hash value;
- Signature;
- SD identifier;
- SCA or SCDev identifier;
- Signer's Certificate Identifier;
- Signature Policy Reference;
- Commitment Type.

Some information, especially the hash value and the electronic signature can and should be inserted by the SCDev in the logging record. The other information, e.g. the Signer's Document identifier, has to be provided by the SCA.

Note: The requirement of a written log record is that it cannot be modified.

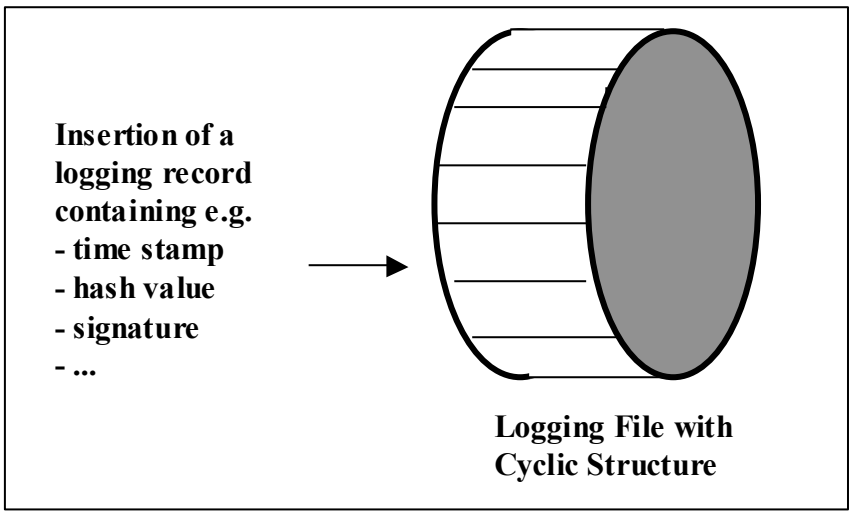

Figure 12 - Signature logging

If signature logging is supported by the SCA and logging takes place on the SCDev , then the application specific SLC component has to organise the writing of a logging record by providing the part of information to the SCDev that is not known by the SCDev . To establish this information, an interaction with the signer may be necessary e.g. for obtaining the Signer's Document reference. Furthermore, the SLC component must be able to retrieve the logging information from the SCDev by an appropriate interaction with the SSA component and be able to display it.

When the logging is performed by the SCA, more information can be logged. In particular it can log actions for all, or for a selected set of the SCA components.

The log function provides the following possibilities:

- to record the most recent signatures generated; and
- to be able to detect any misuse of the SCDev or the SCA.

Annex D (Informative) - SCDev Holder Indicator (SHI)

Certain types of secure signature creation device may not have a suitable place for writing the SCDev holder's name (e.g. PCMCIA tokens, USB tokens and plug-in cards). For these types of SCDev s, it is very useful to have the SCDev holder's name in electronic form, so that it can be retrieved and displayed if necessary by the SCA. SCA implementation should be capable of displaying this information to the Signer over the SIC component.

Annex E (Informative) - References

- ISO/IEC 8613 Information technology – Open Document Architecture (ODA) and interchange format:

The following sources have been consulted in preparation of the Signer's Interface requirements:

- DIN V66291-1: Chipcards with digital signature application/function according to SigG and SigV. Version 1.0 from 15.12.1998, editorial revised 24.04.2000;
- TeleTrusT - Office Identity Card, Version 1.0, 06.07.2000;
- ISO /IEC JTC 1 Business Team on Electronic Commerce Doc. 071;
- ISO TC 68 /CEN TC 224 SC 6 Project Team on Electronic Commerce;
- ITU-T E.135 (10/93) Human Factors aspects of public telecommunications terminals for people with disabilities;
- ITU-T E.121 (07/96) Pictograms, symbols and icons to assist users of the telephone service;
- ITU-T F920 (02/95) Procedures for designing, evaluating and selecting symbols, pictograms and icons;
- ITU.T E.161 (05/95) Arrangement of digits, letters and symbols on telephones and other devices that can be used for gaining access to a telephone network;
- ITU-T F.902 (02/95) Interactive services design guidelines;
- ITU-T E.134 (03/93) Human Factors aspects of public terminals: Generic operating procedures;
- ICTSB PT Report on Design for All and Assistive Technology;
- ICTSB PT Consumer Requirements for ICT;
- EN 1332-Identification Card Systems: Man-Machine Interface:
    - Part One: User Interface dialogue design specifications;
    - Part Two: Tactile identifier;
    - Part Three: Keypads;
    - Part Four: Coding of Special User Requirements;
    - EN 29241-10 Ergonomics requirements for VDU's – Part Ten: "Dialogue Design Principles";
- ES 201 381 Keypads and keyboards for telecommunications equipment;
- ETR 333 Text Telephony; Basic user requirements and recommendations;
- ETR 029 Access to telecommunications for people with special needs; Recommendations for improving telecommunications terminals and services for people with impairments;
- ETR 116 1994 - ISDN Terminal Design;
- ETR 334 The implications of ageing for the design of telephone terminals;
- ETR 345 Characteristics of telephone keypads and keyboards; Requirements of elderly and disabled people;
- E.138 1998 Public terminals for the elderly;
- User Requirements for Smart Card Systems, Balfour A, 1995, Deliverable 3, SATURN project., EU TIDE Project 1040;
- Cost 219 "Proceedings of the Cost 219 seminar on Smart cards and disability", 199
- Access Prohibited? Information to Designers of Public Access Terminals, Royal National Institute for the Blind (UK);

- Access to ATMs: UK Design Guidelines, Centre for Accessible Environments;
- Self service for everyone? – Guidelines for the procurement and installation of self-service systems to meet a Design for All Approach. Delta Centre, Norway, 2000.