CEN

WORKSHOP

AGREEMENT

CWA 14171

May 2004

ICS 03.160; 35.040

Supersedes CWA 14171:2001

English version

General guidelines for electronic signature verification

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 14171:2004 D/E/F

Contents

Foreword

Successful implementation of European Directive 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 general guidelines and recommendations for Electronic Signature Verification. The CWA is intended for use by developers and evaluators of a Signature Verification Application and of its components.

This version of this CWA 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 to purchasers from the CEN Central Secretariat.

This document supersedes CWA 14171:2001.

Introduction

Although there are no formal requirements for signature verification specified in Dir.1999/93/EC[1], Annex IV (the text of which is reproduced in Annex A of this present CWA) recommends that:

"During the signature-verification process it should be ensured with reasonable certainty that:..."

Thus, in order to achieve this "*reasonable certainty",* there is a need for general guidelines on signature verification procedures, including both the products used for verification, and their management.

Signature verification is a process that can be performed in many ways, for example:

- by a natural person, using his workstation and accompanying software to request verification of a received signature,
- by a computer program, using an automated procedure.

Dir.1999/93/EC [1] mentions "data displayed to the verifier", which might be interpreted as verification by a natural person. However, the second case will be useful in electronic commerce, and guidelines are also needed for automated signature verification. Also, the term "displayed" should be interpreted in a more general sense as "presented", since the signed data may be any type of media (text, sound, video etc).The following are the major parties involved in a business transaction supported by electronic signatures:

- Signer,
- Verifier,
- Certification Service Provider,
- Arbitrator.

The **Signer** is the entity which creates the electronic signature.

The **Verifier** is the entity which verifies the electronic signature, it may be a single entity or multiple entities.

The **Certification Service Providers** (CSPs) are one or more service providers which help to build trust relationships between the signer and verifier. They may be used by the signer and verifier to assist them in performing their tasks.

The **Arbitrator** is an entity able to arbitrate disputes between a signer and a verifier.

The signer must provide at least a basic form of Electronic Signature. This basic form does not protect against all potential threats caused by signers' certificates revocation, certificate issuer signing key revocation, signing algorithms and/or keylength weakening, etc., which can undermine the signature reliability. An advantage of this basic form is that it can be created without accessing on-line ancillary services. Moreover, a basic electronic signature may be sufficient in the case of some short-lived transactions, i.e. a signature a party will rely upon within a few hours, before the next useful revocation information is made available. This form is however insufficient to settle disputes in the long term. In order to provide long term verification properties some additional information need to be captured after the electronic signature has been generated. In order to differentiate between these differing circumstances this CWA uses two distinct terms: Initial Verification, and Subsequent Verification.

An **Initial Verification** must be performed within a suitable time after an electronic signature has been generated, in order to capture additional information that will support later verification, potentially over long timescales. The information to be captured and held will relate to the signer's certificate status and validity at the time of the signature and additional information whose capture must be delayed to allow for the 'pipeline' effect of any system processes, e.g. for the promulgation of a revocation decision.

If such data are correctly collected, **Subsequent Verifications** may be successfully performed years after the electronic signature was produced. In order to be able to perform a Subsequent Verification there should be no need to capture more data than those captured at the time of the Initial Verification. Exceptions are, for example, the revocation status of a TSU certificate and additional data acquired for archiving purposes. For

an archive system more data may need to be subsequently captured if the cryptography that was used at the time of the signature is no longer considered to be strong enough to protect the archived data.

This document identifies those data that need to be captured and archived so that they can be later used for arbitration, should a dispute occur between the signer and a verifier. This document also identifies the security requirements for the various elements of a signature verification system.

In order to contribute to the interests of the consumers, i.e. consumer confidence and trust in electronic signatures, the signature verification interface should be as easy to perform and no more difficult to accomplish than is the verification of a hand written signature. It should reduce the probability of human errors and be accessible to most users. This document provides recommendations for the use of the interface and guidance on organisational measures to achieve this confidence.

The present document does not specify how the requirements identified may be assessed by an independent party, nor does it address the requirements for information to be made available to such independent assessors, or requirements upon such assessors. These are addressed in CWA 14172-4 "EESSI Conformity Assessment Guidance: Part 4 - Signature creation applications and general guidance for electronic signature verification".

1. Scope

This document sets out general guidelines on the recommended functionality and assurances for electronic signature verification, in the light of the recommendations in Annex IV from [Dir.1999/93/EC]and in the interest of the consumer.

Its primary purpose is to provide guidance on the way to verify qualified electronic signatures that are equivalent to handwritten signatures according to Article 5.1 of Dir.1999/93/EC [1] , and to complement them with additional data that may help in assessing their validity long after their signing time. Signatures with such additional data have been called “Enhanced Electronic Signatures”.

2. References

[1]. DIRECTIVE 1999/93/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 13 December 1999 on a Community framework for electronic signatures

[2]. ETSI SR 002 176: Electronic Signatures and Infrastructures (ESI) - Algorithms and Parameters for Secure Electronic Signatures

[3]. ETSI TS 101 733: Electronic Signature formats

[4]. ETSI TS 101 903: XML Advanced Electronic Signatures (XAdES)

[5]. ETSI TS 101 456: Policy requirements for Certification Authorities issuing qualified certificates

[6]. ETSI TS 101 862: Qualified Certificate Profile

[7]. ETSI TS 102 231: Requirements for Trust Service Provider status information

[8]. RFC 2560 X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP

[9]. RFC 3280: PKIX Certificate and CRL Profile

[10]. RFC 3161: Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)

[11]. ISO/IEC 9594-8 – Information technology — Open Systems Interconnection — The Directory: Public-key and attribute certificate frameworks

3. Definitions

Certification authority (CA): An authority trusted by one or more users to create and assign certificates. Optionally the certification authority may create the users' keys. – ISO/IEC 9594-8 [11]

Certificate identifier: a unique identifier of a Certificate consisting of the name of the CA and of the certificate serial number assigned by the CA.

Certificate policy: A named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements. ISO/IEC 9594-8 [11]

Certificate validity period: The time interval during which the CA warrants that it will maintain information about the status of the certificate, i.e. publish revocation data. – ISO 9594-8 [11]

Certificate Revocation List: A signed list indicating a set of certificates that are no longer considered valid by the certificate issuer. – ISO 9594-8 [11]

Certification path: A chain of multiple certificates, comprising a certificate of the public key owner (the end entity) signed by one CA, and zero or more additional certificates of CAs signed by other CAs. (RFC 3280 [9])

Certification-service-provider: an entity or a legal or natural person who issues certificates or provides other services related to electronic signatures; [1]

Commitment Type: a signer-selected indication of the exact intent of an electronic signature [3].

CRL distribution point*:* A directory entry or other distribution source for CRLs; a CRL distributed through a CRL distribution point may contain revocation entries for only a subset of the full set of certificates issued by one CA or may contain revocation entries for multiple CAs. [11]

Data To Be signed (DTBS); The complete electronic data to be signed (including both SD and Signature Attributes)

Digital Signature: data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery, e.g. by the recipient (ISO 7498-2).

End entity: A certificate subject which uses its public key for purposes other than signing certificates. – ISO/IEC 9594-8 [11]

Enhanced Electronic Signature: an Electronic Signature complemented with additional data that may help in assessing their validity long after their signing time.

Grace period: time period which permits the certificate revocation information to propagate through the revocation process to relying parties; it is the minimum time period an initial verifier has to wait to allow any authorized entity to request a certificate revocation and the relevant revocation status provider to publish revocation status

Hash function: A function which maps string of bits to fixed-length strings of bits, satisfying the following two properties:

- It is computationally unfeasible to find for a given output an input which maps to this output
- It is computationally unfeasible to find for a given input a second input which maps to the same output

[ISO/IEC 10118-1]

Initial verification: a process performed by a verifier after an electronic signature is generated in order to consolidate the signature by capturing additional information that will support its **Subsequent Validation**.

Object Identifier: a sequence of numbers that uniquely and permanently references an object.

Online Certificate Status Provider: an on line trusted source of certificate status information.

Parallel signatures: the application of separate independent signatures to the same SD

Public key: That key of an entity's asymmetric key pair which can be made public. [ISO/IEC 9798-1]

Private key: That key of an entity's asymmetric key pair which should only be used by that entity [ISO/IEC 9798-1]

Qualified certificate: a certificate which meets the requirements laid down in Annex I of Dir.1999/93/EC and is provided by a certification-service-provider who fulfils the requirements laid down in Annex II of the Directive; [1]

Qualified electronic signature; an advanced electronic signature which is based on a qualified certificate and which is created by a secure-signature-creation device (Ref. Art. 5.1 of Dir.1999/93/EC [1])

Relying party: A user or agent that relies on the data in a certificate in making decisions. ISO/IEC 9594-8 [11]

Signature attributes: Additional information that is signed together with the SD.

Note: in ETSI TS 101 903 [4], "signed attributes" are called "signed properties".

Signature Policy: a set of technical and procedural requirements for the creation and verification of an electronic signature, under which the signature can be determined to be valid.

Signature Policy identifier: Object Identifier that unambiguously identifies a Signature Policy.

Signature Policy issuer: An organisation that creates, maintains and publishes a signature policy.

Signature Policy Issuer name: A name of a Signature Policy Issuer.

Signature verification: a process performed by a verifier after the creation of an electronic signature to determine if an electronic signature is valid.

Signature-verification-data: data, such as codes or public cryptographic keys, which are used for the purpose of verifying an electronic signature; [1]

Signature-verification device: configured software or hardware used to implement the signature-verification-data; [1]

Signer role: the identification of an organisational/operational function that a signer is claiming or allowed to have when invoking the signature process.

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

Subsequent Verification: a process performed by a verifier some time after the initial application of a signature, to assess a signature's validity, based on the data collected at Initial Verification time.

Time-Mark: A proof-of-existence for a datum at a particular point in time, in the form of a record in a secure audit trail, which includes at least a trustworthy time value and a hash representation of the datum.

Time Stamp Token: A proof-of-existence for a datum at a particular point in time, in the form of a data structure signed by a Time Stamping Authority.

Time Stamping Authority: An authority trusted by one or more users to provide a Time Stamping Service.

Time Stamping Service: A service that provides a trusted association between a datum and a particular point in time, in order to establish reliable evidence indicating the time at which the datum existed.

Time Stamping Unit: set of hardware and software which is managed as a unit and has a single time-stamp token signing key active at a time

Validation data: additional data, collected by the signer and/or a verifier, needed to verify the electronic signature. It may include: certificates, revocation status information, time-stamps or Time-Marks.

Verifier: an entity that validates or verifies an electronic signature. This may be either a relying party or a third party interested in the validity of an electronic signature.

What Is Presented is What Is Signed (WIPIWIS): a description of the required qualities of the interface able to unambiguously present the SD to the verifier according to the SD format.

4. Abbreviations

CA	Certification Authority
CSP	Certification Service Provider
CP	Certificate Policy
CRL	Certificate Revocation List
ES	Electronic Signature
OID	Object IDentifier
OCSP	Online Certificate Status Protocol
PDA	Personal Digital Assistant
QC	Qualified Certificate
RA	Registration Authority
SCA	Signature Creation Application
SCS	Signature Creation System
SSCD	Secure Signature Creation Device
SD	Signer’s document
TSA	Time Stamping Authority
TSP	Trust Service Provider
TSS	Time Stamping Service
TSU	Time Stamping Unit
WIPIWIS	What Is Presented Is What Is Signed.

5. Verification processes

5.1 Signature lifetime

Verification requirements, and consequently the data to be associated to a signature, vary depending on the foreseen signature life. The following cases can be outlined to better define the verification processes:

1. **ephemeral signature:** signatures that need not to be kept beyond the next meaningful revocation information issue time;
2. **short term signatures:** signatures that are to be verified for a period of time that does not go beyond the signers' certificate expiration date.
3. **long term signatures:** signatures that are expected to be verified beyond the signers' certificate expiration date and, possibly, even after the expiration date of the certificate of the signers' certificate-issuing CA.

Only verification of short and long term signatures is covered in this document.

5.2 Initial and subsequent verification

The term **verification** is used where an electronic signature is determined to be valid or not. Two specific instances of verifications, Initial verification and Subsequent Verification are used in this document (and are defined in Section 3):

The collection of revocation data suitable to create an Enhanced Electronic Signature, as defined in section 3 can only be done by a person acting as the Initial Verifier, because those data must be collected within a short period after the signing time. If this person is the signer itself, in this case he/she is acting as a Verifier.

It is also intuitive that the signer is likely to have an interest in adding time reference to the signature to prevent his/her signature to become void should any certificate in the path be revoked. Similarly, the Initial Verifier could add this time reference, in case the signer did not do it.

A Subsequent Verification should collect additional data in the following three cases:

1. if a Time Stamp Token has been used to assert a moment before which the signature existed, and the certificate of the TSU which issued the TST is about to expire, an additional TST might be needed to extend the signature validity;
2. if a Time Stamp Token has been used, revocation information on the relevant TSU certificate should be collected too; if the TSU certificate status is "revoked", the TST should not be deemed as valid;
3. if the cryptography that was previously used is likely to be broken soon, additional information needs to be gathered in order to extend the life-time of the signature.

These remarks apply equally to both advanced and qualified electronic signatures.

5.3 Verification information requirements

To assess the validity of a long term electronic signature, the basic information to ascertain, in addition to the signatures' intrinsic cryptographic validity, is whether the signer's certificate was valid at signing time.

The certificate validity may be affected by three factors:

1. the compromise of any of the signing keys along the certificate path, starting from the signer's certificate up to that of the authority which is one of the trust anchors acceptable by the verifier, for example in an explicit signature policy;

2. the weakening of the algorithms or key-lengths used to apply the signatures (any applicable signature: the one applied to the SD or the signature to any certificate in the path), to the extent that one of them may be broken in a reasonable time;

3. organisational reasons, such as changed affiliation or superseded certificate.

Once it turns out that any of the above mentioned keys has been compromised, or that the involved algorithm is no longer reliable, or that the certificate was revoked for organisational reasons, the point is to evaluate if the certificate invalidation occurred before or after the signing time. Hence the need to have some kind of time reference for the signature along with timely information on the status of each certificate relevant to the signature, from the relevant trust service provider.

The basic data that are to be added to a signature for a verifier to assess its validity in the long term are, consequentially:

- time related information captured as soon as possible after the signing time, to state that the signature was issued before the time indicated;

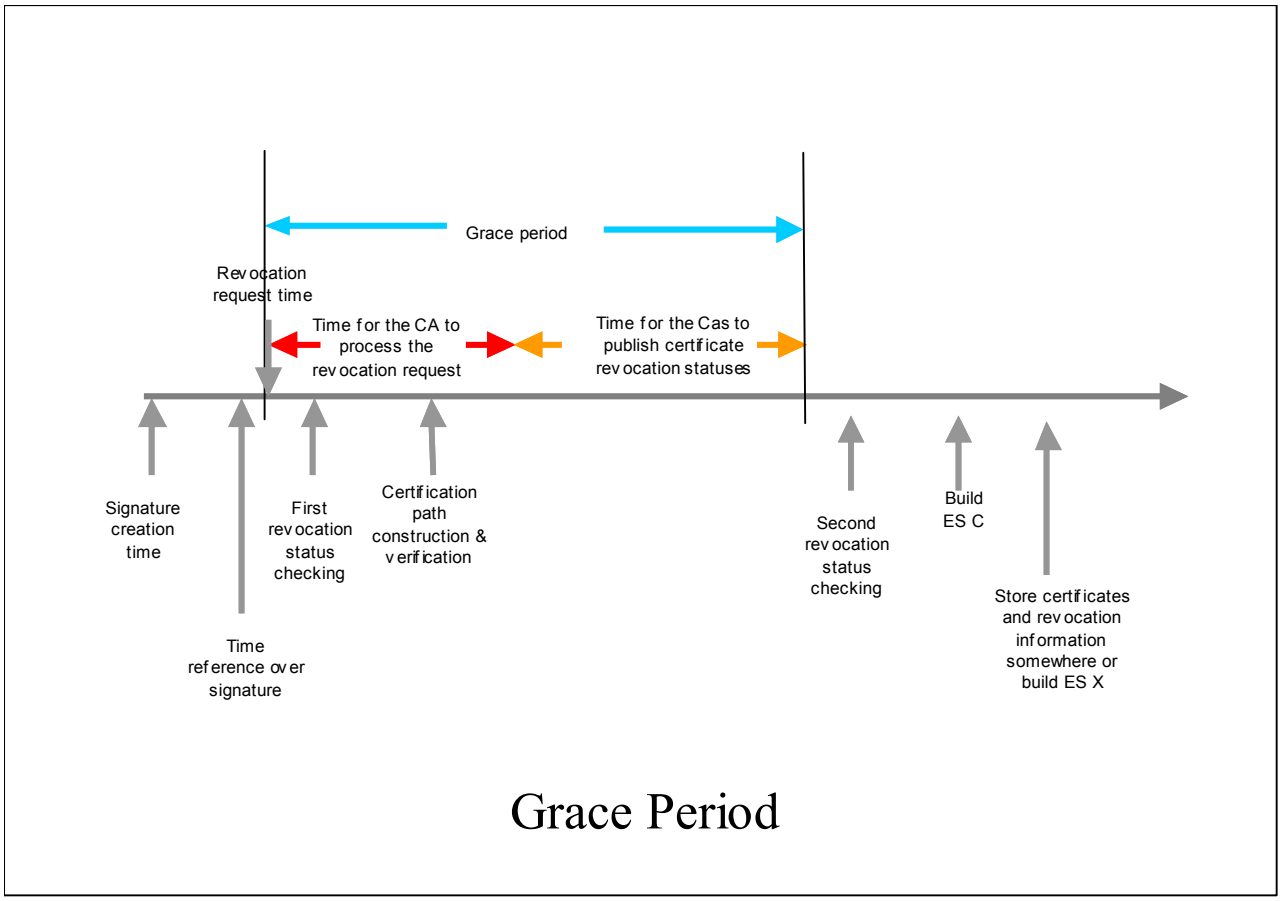
- revocation status information of the certificates involved in the path, to be captured after a "grace period".

The "grace period" is the minimum time period during which an initial verifier has to wait to allow a signer or any other authorized entity to request a certificate revocation and the relevant revocation status provider to publish revocation status.

The relying party, before assessing the validity of a certificate associated to the signature, should ascertain that at least the grace period has elapsed since a signature relevant time. This signature relevant time should be the time indicated in an associated TST or in an associated time mark. Lacking any of these two time references, a verifier can rely on the time the signed object was first received.

To compute the grace period start time, relying parties should rely on the revocationTime as specified in the CRL for the involved certificate.

A relying party may also choose to take into account the invalidityDate or any other information that reports an attestation by the signer of a point in time, earlier than the revocationDate, where the signer states that the associated private key was no more to be trusted as uncompromised. Relying parties, on the other hand, must be aware that, while the revocationDate moment is attested by the CA, the invalidityDate has no such endorsement, therefore this choice is at their own risk.



It is up to any single party to add the required information to the signature, depending on his/her interest in the specific case.

Other information, ancillary to the previous ones, can also be added by any of the involved parties, depending on their specific interest.

5.3.1 Time related information

In order to determine unambiguously whether the signer's certificate and the attribute certificates, if any, were *valid at the time the signature was generated*, time related information can be used to establish a time before which a signature was created. It can be of two different types:

a) a **Time Stamp Token** from a Time Stamping Authority (TSA), or

b) **a Time Mark,** based on a secure audit trail, in which are recorded, as a minimum, a trusted time and a value uniquely linked to the electronic signature.

The first case has several advantages: it allows prooving that the signature was generated before the time indicated in the time stamp without revealing any other information; it only uses digital information that can be copied from one media to another without losing any of the initial properties.

The second case requires specific procedural requirements to avoid disclosing, at the time of verification, potentially sensitive information such as the format of the audit trail, the procedures uses to create the audit trail, the physical media used to support the audit trail at the time it was recorded, and the other records that are in the audit trail. It would be impossible to standardize the format of such audit trails as well as the use of some physical media adequate for such a recording. In addition, the Trust Service Provider SHOULD be independent from the parties, to ensure no collusion between one of the parties and the Trust Service Provider. For all these reasons, the use of time stamps, although not mandatory, is to be preferred.

However, these time related methods provide no information about when precisely the signature was generated

5.3.2 Certificates and revocation status information

5.3.2.1 Linking signature and certificate set

Linking the signatures to the certificate actually used by the signer prevents its substitution and thus protects from attacks like assigning the signature to a different signer or signer's role.

Similarly, adding to the signatures the certificates of the issuing CA and of the hierarchically higher level CAs, may help to build a certification path, which in turn may be helpful to identify the status of these CAs.

This link may be implemented by inserting into the signature only a reference to such certificates or by explicitly including the certificates themselves. The reason for each different usage is detailed below.

It is to be noted that a unique and unmistakable reference to the signer's certificate is of paramount importance for the signature integrity.

Note: This avoids actual or purported certificate substitutions with other certificates with different semantics, regardless if they exist at signing time or are issued later on, possibly with a compromised signing key from the relevant CA.

Even when the signer's different certificates are related to different signature verification data it is useful to specify this piece of information, because it helps the verifier identify the correct signature verification data.

5.3.2.2 Linking signature and certificate revocation status.

A CA is perfectly free to delete the reference to a revoked certificate from a CRL upon certificate expiration. In this case, if the verification is performed after the certificate expiration date, it would not be possible to ascertain if this certificate was revoked when a related signature was issued. Hence the need to include in signatures, that are supposed to last beyond the certificate expiration date, a reference to suitable information on the certificate status at the moment of the signature.

Similarly, since keys, used by certificate status information providers to sign such information, may be revoked and will always expire, it is necessary to include in a signature this information in its entirety and captured when the status signing key is still valid, to prevent attacks like substitution of an archived CRL that was only referenced, but not included, in a signature.

5.4 Signature formats as specified in TS 101 733 and in TS 101 903

A paradigm for this document specifications is the set of ES formats described in TS 101 733 [3], where the following electronic signature formats are provided for.

- **Electronic Signature (BES or EPES[1])**, which includes the digital signature and other basic information provided by the signer. The ES satisfies the requirements for advanced electronic signatures as defined in the European Directive [1]. It provides basic authentication and integrity protection and can be created without accessing on-line (time stamping) services. However, without the ability to position the electronic signature in time, the digital signature does not protect against the risk that an electronic signature, issued when the corresponding certificate was valid, is deprived of validity if the same certificate is later on revoked;

---

[1] A **Basic Electronic Signature** (BES) contains the SD, a collection of mandatory signed attributes, the digital signature value computed on the SD and on the signed attributes, when present. It may also contain a collection of additional signed attributes and a collection of optional unsigned attributes.

An **Explicit Policy based Electronic Signature** (EPES) incorporates in addition a signed attribute, protected by the signature, which indicates that a signature policy is mandatory to use to validate the signature and specifies explicitly that signature policy. The signature may also have other signed attributes required to conform to the mandated signature policy.

- **ES with Time (ES-T),** in which either a **Time Stamp** from a Time Stamping Authority is added to the Electronic Signature, or a TSP is requested to create and keep in a secure audit trail a **Time Mark** related to the Electronic Signature; this is an initial step towards providing long term validity because it helps evaluate if the signature was issued before or after a possible certificate revocation;

- **ES with Complete validation data (ES-C**), with which the references to (but not the values of) the complete set of data supporting the validity of the electronic signature (e.g. certification path and revocation status information) are added to the ES-T. The ES-C thus contains both the references of the validation data *and their hash values*. This allows making sure that the revocation status information which were actually captured are the referenced ones. The certificate and revocation status information hash values prevent fake certificates and revocation status information from being produced, upon compromise of the CA key. The complete set of data supporting the validity of the electronic signature does not necessarily need to be kept together with the Electronic Signature but may be kept somewhere else. The ES-C is the common denominator of two other forms of ES. One form (identical to the ES-C) allows to store these values elsewhere, e.g. in some central storage, while the other form (ES-X) allows to store all the values of the validation data together with the ES;

- **ES-C with complete certificate and certificate status information (ES-X**), with which ES-C are extended with validation data that include the values of the certificates and certificate status information.

- **ES-C or ES-X time stamped for archive (ES-A)**. Before the algorithms, keys and other cryptographic data used at the time the ES-C was built become weak and the cryptographic functions become vulnerable, or the certificates supporting previous time-stamps expires, the signed data, the ES-C and any additional information (ES-X) should be time-stamped. If possible this should use stronger algorithms (or longer key lengths) than in the original time-stamp. This additional data and time-stamp is called Archive Validation Data. (ES-A). The Time-stamping process may be repeated every time the protection used to time-stamp a previous ES-A becomes weak. An ES-A may thus bear multiple embedded time stamps.

In the case of TS 101 733 the signer SHALL provide at least the BES or the EPES form.

An XML version of TS 101 733 has been published as TS 101 903 [4], where the correspondent XML versions of ES, ES-T, ES-C, ES-X, ES-A are detailed as XAdES, XAdES-T, XAdES-C, XAdES-X, XAdES-A.

5.5 Initial Verification inputs

The Initial Verification of an electronic signature requires:

- the Signer's Document, which is the document that is signed by the signer;

- the electronic signature, over the Signer's Document, that, when the TS 101 733 is used as reference, shall comply with the suitable format among those specified in TS 101 733[3];

- a set of explicit or implicit rules to follow for the verification process, for example which trust anchor (root certificate) to use for verification. These rules may be formalized and contained in a signature policy.

A Signature Policy is set of technical and procedural requirements for the creation and verification of an electronic signature, under which the signature can be determined to be valid. For further discussions on the concept of signature policies, please refer to Annex D.

Optionally, the other additional data associated with the electronic signature, specified in Section 5.3, may be present.

The digital signature, around which the electronic signature is built, is applied over the following elements:

- the Signer's Document or a hash of it (i.e. a document digest),

- an unambiguous reference to the signer's certificate selected by the signer, e.g. the certificate itself or a reference to it together with a hash value of the certificate.

In addition, the digital signature may cover the following optional information, if present:

- the identifier of the referenced signature policy;
- a Data Content type attribute (e.g. Contents-hints as specified in TS 101 733 [3] or DataObjectFormat as specified in TS 101 903 [4]) that identifies the format of the Signer's Document (when electronic signatures are not exchanged in a restricted context) to help the verifying SW application present the relying party with the Signer's Document (text, sound or video) in exactly the same way as intended by the signer;
- the signing time, as claimed by the signer or
- a time stamp token linked to the Signer's Document (to prove that the electronic signature was performed after the time it specifies);
- the claimed or certified role(s) assumed by the signer in creating the signature;
- the commitment-type-indication attribute which illustrates a type of commitment on behalf of the signer;
- the location of the signer, as claimed by the signer, at which the signature was created.

5.6 Initial verification outputs

The **initial verification process** validates an electronic signature.

Where used, the requirements of the signature policy must also be taken into account.

There are two outputs: an output status and output data, called validation data.

5.6.1 Output status

The output status of the initial verification process can be:

- passed verification;
- failed verification;
- incomplete verification.

A **Passed Verification** response indicates that the signature has been found to be valid.

A **Failed Verification** response may depend on several factors, e.g.: it may indicate that the signature format is incorrect, or the digital signature value failed verification or the signer's certificate has been revoked.

An **Incomplete Verification** response indicates that the format and digital signature verifications have not failed but there is insufficient information to determine if the electronic signature is valid. This would for example be the case when a time stamped signature is verified before any possible revocation in progress has been published. It may be necessary that the electronic signature be verified again at a later time when additional validation information might become available. Also, in the case of incomplete verification, additional information may be made available to the application or user, thus allowing the application or user to decide what to do with partially correct electronic signatures.

Note: A Failed Verification or Incomplete Verification may be caused by the fact that the signer's certificate has been revoked with the reason code certificateHold, which means that the certificate has been suspended and is "on hold". In such a case, the initial verifier may want to perform a new initial verification at a later time, with the possibility of achieving a Passed Verification.

5.6.2 Validation Data

The **Validation Data** is collected by the verifier if one of the Enhanced Electronic Signature formats is to be produced. If signers collect these data themselves, they are actually acting as verifiers, since they have already completed the strict signing process and are performing a new one: the verification one.

Where an explicit signature policy is indicated, its requirements shall also be met.

Basically, if there is a need for a Subsequent Verification, the validation data should contain the proof that the certification path that was used was valid *at the time the signature was generated*.

As stated in section 5.3, before collecting the validation data, two questions should be answered:

1. Which is the presumed signature lifetime? If it does not go beyond the signer's certificate's expiration date, only a time stamp token or a time mark may be necessary, together with the reference to (or the value of) of the signers' certificate, that is the values of the certification path and the associated revocation status need not being stored by the verifier.

2. Who is responsible for collecting these data? The signers have completed their task; the initial verifier may be happy with the received result. Only verifiers who are also the signature "keepers" and who therefore know they will need the signature in the future will have interest in building the ES-C format.

If it is expected that the signature lifetime will go beyond the signers' certificates' expiration date, thus being a long term signature, the following requirements apply.

Based on the presumed signature "lifetime", the verifier should define to which extent a certification path and the associated revocation status information for each certificate from the path should be captured. The verifier must also decide which type of validation data to include: whether just the reference to the CA certificates and to the revocation status information, or the complete certificates and certificate revocation status information, be they CRLs or certificate status information provided by an on-line service.

In order to prove that the data was captured before revocation occurred or before the end of the validity period of the certificates, and if there is a need for a Subsequent Verification, a time stamp over the electronic signature or a secure audit trail should be used to provide evidence of the timing of given events. It is required, as a minimum, that either the signer or verifier obtains a time stamp over the signer's signature, or that a time mark is kept by a trusted service provider.

Referring to the TS 101 733 [3] and TS 101 903 [4] formats, mentioned in section 5.4, the signer may decide, in some cases, to provide more data than the ES form and in the extreme case could provide an electronic signature with complete validation data (e.g. the ES-C form). The **Validation Data** *may* thus also be collected by the signer and fully provided to the verifier. In this case the signer acts as the Initial Verifier and abides by the requirements for capturing the Validation Data at the right time, which would imply that the signer retains the produced signature until the "suitable revocation status information" becomes available. It is to be noted that in this case the Initial (where applicable) or Subsequent Verifier should ascertain that a "suitable time" has elapsed before adding the revocation status information, since the signer might have interest to create an ES-C with a still unreliable set of revocation information.

When there is a need for a Subsequent Verification, if the signer does not provide ES-T, the verifier may create the ES-T on first receipt of an electronic signature. The verifier then may create the ES-C when a suitable complete set of revocation and other validation data is available.

As previously hinted at in item 2 of section 5.2, if a TST has been used, a subsequent verifier should verify if the relevant TSU certificate was valid at generation time by means of appropriate methods (e.g. CRLs or OCSP Response). It can be remarked that referencing or including in a signature this TSU certificate revocation status information captured at the time of the signature would not per se add sufficient trust to the ES. In fact, should a TSU signing key be compromised, the time specified in a TST signed with that key should not be trusted.

5.6.3 Extended forms of validation data

A verifier may want to keep information that the certification path and the revocation information used at the time of the signature were valid, even in the case where one of the issuing keys or status information signing keys is later compromised. Additional information in the validation data is able to provide such a protection.

The complete validation data (ES-C) described above may be extended to form an ES with eXtended validation data (ES-X) to allow the storage of all the values of the validation data together with the ES. This means in particular:

- the signer's certificate,

- all the CA certificates that make up the full certification path, as referenced in the ES-C, and

- all the associated revocation status information, as referenced in the ES-C.

This form of ES with extended validation data is called ES-X.

Alternatively, it is possible to keep at least the signer's certificate in the ES-X and store CAs certificates and CRLs at a Trust Service Provider. This is more efficient in terms of storage, for example when electronic signatures are received from subjects certified by the same CAs.

It is worth noting that during subsequent verifications it may be necessary, before the algorithms, keys and other cryptographic data used at the time the validation data was constructed become weak or the certificates supporting previous time stamps expires, to time stamp the signed data and the validation data using stronger algorithms (or longer key lengths) than in the original time stamp token. The Time stamping process should be repeated every time the protection used to time stamp previous validation data becomes weak. Validation data may thus include multiple embedded time stamps.

5.7 Verification process rules

5.7.1 Signer Certificate

5.7.1.1 Verification of Electronic Signatures using Qualified Certificates

For the verification of Electronic Signatures using Qualified Certificates, the signer’s certificate must be a Qualified Certificate. The use of an SSCD is not necessarily required.

According to ETSI TS 101 862 [6], section 4.2.1, the specification that a certificate is issued as a Qualified Certificate can be provided either:

1. by indicating in its Certificate Policies extension a reference to a certificate policy that clearly expresses that the certificate has been intentionally issued as a Qualified Certificate and that the issuer claims compliance with the Directive (Certificate Policies of this kind are provided in TS 101 456 [5]); or

2. by using the Qualified Certificate Statements extension as defined in TS 101 862.

A combination of both techniques is permitted.

5.7.1.2 Verification of Qualified Electronic Signatures

For the verification of Qualified Electronic Signatures, the signer’s certificate must be a Qualified Certificate, as detailed in the previous section. In addition, the use of an SSCD is also a requirement on the signer.

To achieve this, the certificate policy identified in the Certificate Policies extension, as defined in section 4.2.1.5 from RFC 3280 [9], must clearly state that the certificate was issued as a Qualified Certificate to a subject who is bound to use an SSCD. This can be done either by referencing the policy "**QCP public + SSCD"** as defined in ETSI TS 010 456 [5] or by referencing a CSP-specific policy that clearly expresses the required use of an SSCD.

5.7.1.3 Verification of other electronic signatures

For the verification of other electronic signatures there is no requirement for the signer's certificate to be a Qualified Certificate.

5.7.2 Rules for Certification path construction/verification

When using an Internet X.509 Public Key Infrastructure, section 6 of RFC 3280 [9] extensively describes an algorithm for validating certification paths which is very detailed and is therefore recommended to be used as a reference. It should be noted that subsection 6.1 of the RFC states:

“This text describes an algorithm for X.509 path processing. A conformant implementation MUST include an X.509 path processing procedure that is

```
functionally equivalent to the external behaviour of this algorithm.
However, support for some of the certificate extensions processed in this
algorithm are OPTIONAL for compliant implementations.  Clients that do not
support these extensions MAY omit the corresponding steps in the path
validation algorithm.
```

It is also worth noting that no reference is made in Section 6 of RFC 3280 to online protocols to provide information on certificate status, like OCSP which is detailed in RFC 2560 [8]. These online services actually provide signed objects for which, consequently, a certificate path is to be processed.

5.7.3 Rules for the use of Revocation Status information

The verifier should take into account information in the certificate in deciding how best to obtain and check the revocation status (e.g. a certificate extension field about authority information access or a CRL distribution point).

Before assessing that an electronic signature is valid, the verifier should ascertain that at signing time:

I. the used certificate was within its validity period;

II. the certificate used was not revoked. However, there is an inevitable delay between a possible request for revocation, and a report of the corresponding revocation being distributed. To allow greater confidence in the validity of a signature, a "grace period" is necessary, as defined in section 3 and described in section 5.3

The verifier has to pay special attention to the following situation:

1. At “time 1” the electronic signature is applied to the document to be signed. The signer may optionally add to the signed data the time he/she claims the signature was computed, which *per se* cannot be relied upon since the verifier has no possibility to verify it.
2. The electronic signature is Time Stamped or Time Marked at time 2, later than time 1.
3. The grace period is added to time 2 and thus ends at “time 3”.
4. The certificate is revoked at “time 4” which may be earlier or later than time 3.
5. The electronic signature is initially verified at “time 5”, which should not be earlier than time 3.

During verification at time 5 three cases may occur:

(a) no revocation has been issued for the certificate;

(b) a revocation has been issued with a revocation time 4 which is later than time 3; the revocation was issued after the grace period.

(c) a revocation has been issued with a revocation time 4 which is earlier than time 3.

In case (c), the electronic signature fails verification.

In cases (a) and (b) the electronic signature passes verification.

In case that the verifier has no evidence that the electronic signature was time stamped or time marked, then the verifier may have as a time reference for “Time 2” the moment the signature has been received. If the signer’s certificate appears as having been revoked before that time, the electronic signature does not pass verification successfully.

5.7.4 Rules for use of Time-stamping or Time-marking

When time marking is being used, a unique reference to the ES is recorded in a secure audit trail, for example the signature itself or a hash of it, together with a secure recording of the current time.

When time stamping is being used, the following rules should be followed when specifying constraints on the certificate paths for time-stamping authorities and on the time-stamping authority names, general timing constraints, and on TSU certificate status verification.

Refer to RFC 3161 [10] Section 4 to address Time Stamp Token security related matter.

5.7.4.1 Trust points and Certificate paths

Signature keys from time-stamping authorities will need to be supported by a certification path. The certification path used for time-stamping units requires a trust point and possibly path constraints in the same way as the certificate path for the signer's key.

Similar rationales apply to TSU names.

5.7.4.2 Timing Constraints

There will be some delay between the time that a signature is created and the time the signer's signature is time-stamped. However, the longer this elapsed period, the greater the risk of the signature being invalidated due to revocation of its certificate by the signer or by any authorised party.

If there is a need to verify an electronic signature before the end of the certificate validity period, then it is necessary to make use of time stamping or time marking, in order to be able to discriminate between signatures performed before or after the revocation of an element of the certification path, including the signer's certificate.

If there is a need to verify an electronic signature beyond the end of the certificate validity period, then it is necessary to make use of time stamping or time marking before the end of the certificate validity period, and to capture the revocation status information while it is still managed by the CA[2].

5.7.4.3 TSU certificate status checking

The verifier should ascertain that the last valid TST is not revoked at verification time. Please refer to item 2 in section 5.2 and to section 5.6.2 for additional rationale.

5.7.5 Verification of qualified certificate issuer status

Art. 3.3 of Dir. /93/EC [1] requires that:

> "Each Member State shall ensure the establishment of an appropriate system that allows for supervision of certification service-providers which are established on its territory and issue qualified certificates to the public.

Additionally, Art. 3.2 of the said Directive refers to the Certification Service Provider approval:

> "… Member States may introduce or maintain voluntary approval schemes aiming at enhanced levels of certification-service provision. …"

In either case, a verifier may want to assess whether the qualified certificates he/she is assessing were issued by a CA that, at the time of issue, was approved by the relevant supervisory scheme of the country it which it was established or approved by any approval scheme.

ETSI TS 102 231 [7] describes the format of a Trust-service Status List (TSL) which might be maintained by any such scheme operator, and in which there might be found status information relating to the TSP which issued the certificate, or any other verification-relevant credentials (e.g. a time-stamp). A verifier should, according to guidance given in [7], determine whether there is available a TSL holding status information pertinent to the TSP in question, and then use that information to establish the status of the certificate-issuing TSP at the time of the act of signing, and by implication any inferences as to the status of the certificate.

---

[2] *"The certificate validity period is the time interval during which the CA warrants that it will maintain information about the status of the certificate"* (ISO 9594-8 section 7).

5.7.6 Rules for algorithm constraints and key lengths

Standards or national law and regulations may identify a set of signing algorithms (hashing, public key, combinations) and minimum key lengths that may be used:

- by the signer when creating the signature;
- in end entity public key Certificates;
- in CA Certificates;
- in Attribute Certificates;
- by the Time Stamping Authority.

A set of algorithms and parameters that may be used for qualified electronic signatures is specified in the SR 002 176 [2].

5.7.7 Rules for use of signer roles

Signer roles are optional and can be supported as claimed signer roles or as certified signer roles in Public Key Certificates or Attribute Certificates. Attribute Certificates are supported through signed attributes (in TS 101 733 [3]) or through signed properties (in TS 101 903 [4]).

5.7.7.1 Attribute values

The acceptable role attribute types or values may be dependent on the signing context or on signature policies, where used. For example, a user may have several signer roles that allow the user to sign data that imply commitments based on one or more of his roles.

For machine processable verification of attributes, it is crucial that the signer roles are encoded in an unambiguous manner.

5.7.7.2 Trust points for Certified Attributes

When certified attributes are used, they can be certified either within the public key certificate by a certification authority, or by an Attribute Authority that includes them in Attribute Certificates it issues.

In the case of Attribute Certificates, the Attribute Authority needs to be validated as part of the overall verification process of the electronic signature. The trust points for Attribute Authorities do not need to be the same as the trust points used to evaluate a certificate from the CA of the signer. Thus the trust point for verifying roles needs not be the same as the trust point used to validate the certificate path of the user's key.

Attribute Authorities are certified by CAs and are thus are considered as leaves of a certification tree structure.

Naming and other constraints may be required on attribute certificate paths, similarly to other electronic signature certificate paths, but these constraints need not be exactly the same on the AA and CA.

When a signature policy is used, it may identify trust points that can be used for Attribute Authorities (AAs), either by reference to the same trust points as used for Certification Authorities, or by an independent list.

Moreover, it may identify additional constraints on some parameters used as input to the certificate path processing, such as:

- acceptable certificate policies, including requirements for explicit certificate policy indication and whether certificate policy mapping is allowed;
- naming constraints in terms of constrained and excluded naming sub-trees;
- restrictions on the certificate path length.
- other Signature Policy rules.

5.8 Subsequent Verification inputs

The Subsequent Verification may take place even years after the electronic signature was done.

When a signature policy is used, a verifier should unambiguously identify the signature policy that has been referenced either explicitly or implicitly, possibly also by means of the type of data being signed. In this case he should obtain a copy of the signature policy in a trusted way and make sure that what has been received matches the intended signature policy. This is the signature policy to be used for the verification process.

The same controls, already described for the Initial Verification apply, except that the validation data must be already available to the verifier. However, the verifier must ensure that the last valid Time Stamp Token has not been revoked, and may also require the signature to be time stamped (or time marked) again to further extend its validity.

6. Signature verification systems

The security of a signature verification system depends on:

1. its secure development;
2. its correct installation;
3. the availability of ways to prevent or, at least, to detect successful tampering attempts.

Therefore its day to day use relies on the procedure of the installation phase, on some means to make sure that this phase was correctly completed and to detect any security relevant system modification applied since the system was installed.

6.1 Initial Verification systems

An initial signature verification system is composed of:

- the secure signature verification process,
- an interface to enter the Signed Document and to select the electronic signature to be verified (there may be more than one electronic signature attached with the user data),
- an interface for the current time,
- an interface for verification rules to be followed (e.g. a signature policy),
- a display/sound/video interface to present (e.g. display, listen to or visualize) the signed user data with the right format,
- an interface to get the signer's information and the output status after signature verification,
- an optional interface to write in a secure audit trail from an independent Trusted Third Party;
- a network interface to optionally fetch information produced by Trust Service Providers when not provided by the signer (e.g. CA repositories, CRLs repositories, OCSP responders, Time Stamping Authorities);

In Figure 1 the previous components of the Initial Verification process are illustrated.

Additionally the following components can be included in the process:

- where necessary an interface to obtain any status information from available TSLs;
- where applicable an optional interface to get the definition of the Signature Policy.

Error! Objects cannot be created from editing field codes.
Figure 1

When a Time-Mark, instead of a Time Stamp is used, then, one of the Trust Service Providers must be in charge of keeping the electronic signature in a secure audit trail.

6.2 Subsequent Verification systems

A subsequent signature verification system is composed of:

- the secure signature verification process,
- an interface to enter the signer's document and to select the electronic signature to be validated (there may be more than one electronic signature attached with the user data),

- an interface for the current time,
- an interface for verification rules to be followed (e.g. a signature policy),
- a display/sound/video interface to present (e.g. display, listen to or visualize) the signed user data with the right format,
- an interface to get the signer information and the output status after the initial signature verification,
- an optional interface to fetch the recording time of the electronic signature from the secure audit trail of an independent Trusted Third Party.

In the following Figure 2 the previous component of the Subsequent Verification process are sketched.

Additionally the following components can be included in the process:

- where necessary an interface to obtain any status information from available TSLs;
- where necessary an interface to fetch the TST issuing TSU certificate status;
- where applicable an optional interface to get the definition of the Signature Policy.

When Time Marks, instead of a Time Stamps are used, then an interface to ask for an audit trail in order to determine the signing time using a process which is outside the perimeter of the initial signature verification system.

Error! Objects cannot be created from editing field codes.

Figure 2

6.3 Human verification

Humans will use various forms of implementations corresponding to the two generic diagrams presented before in Figures 1 and 2. The implementations will consist of several hardware components supporting software components.

Various user interfaces need to be present.

6.3.1 Selection of electronic signature for verification

The signature verification system has to provide the user with a way to interact with it. When more than one signature are affixed to the signed data, this starts by indicating the number of signatures that possibly exist and offering which one has to be verified.

6.3.2 Presenting the signer's document

The user interface has to present the content of the signer's document in an appropriate way, so that a person verifying a signature should also be able to identify the contents of the signer's document to an adequate extent. The signer's document may be text, voice and video, in many different formats. In order to select the right way to read, listen or visualise the signed user data, the presentation format of the data is always indicated within the electronic signature and that information is thus protected by the digital signature from the signer. This interface has thus to fulfil the requirement "**W**hat **I**s **P**resented **I**s **W**hat **I**s **S**igned" (**WIPIWIS**). If, for any reason, the content of the signed document cannot be presented in exactly the right way, then the user interface should clearly report this.

In a few formats, coded, or macro, instructions may be imbedded in the document in a way that the verifier is presented something different from what was signed without affecting the cryptographic/algorithmic validity of the signature (e.g.: Microsoft® Word, Microsoft® Excel, HTML, etc.). Therefore the user interface should issue a warning to the verifier that the document format may bear this kind of code. For this type of format it is recommended to use, where available, viewers which are able to notify the verifier when something has changed in the SD since it was signed and, where possible, which changes have occurred.

6.3.3 Presenting signer information and output status

The interface should be able to display the purported signer's identity. Since different CAs may unknowingly issue certificates to homonyms or to different persons choosing the same pseudonym, the name of the signer as presented to a verifier may be insufficient to distinguish between several persons with the same name or pseudonym. A certificate owner's name is only meaningful within the issuing CA domain. So it may be also necessary to uniquely identify the chain of CAs up to a trust point."

Thus, in order to define the signer's identity, the verifier may need to escalate the naming hierarchy by going through the following steps that the user interface should support up to when a sufficient reliability is achieved:

1. inspect the subject's Distinguished Name in the signer's certificate;
2. inspect the Distinguished Name of the issuing CA;
3. inspect the Distinguished Name of the hierarchically superior CAs up to a root that is acceptable for the verifier.

Note: the subjectAltName should be inspected, even if the subject's DN is not empty, because if present it may convey useful information on the subject's identity and/or role.

The following information should be presentable at the wish of the verifier:

- the name or the pseudonym of the purported signer specified in the certificate associated with the name of the CSP which has issued the Qualified Certificate (if not done before),
- the claimed date and time of the electronic signature, when present.

Using the interactive interface the user should be able to get additional information like:

- other content of the signer certificate,
- the date and time of the time stamp, or of the associated time mark, if present,
- the place of the signature if present,
- the commitment type under that signature if present,
- the identification of the signature policy or its content, if present,
- rules used in the verification process, in particular trust anchors,
- the claimed role or certified role under which the signature was generated, if present.

Using the interactive interface the user should be able to get the status of the verification process. The output status should be presented in a reliable, unambiguous and non-manipulated manner.

The output status of the initial verification process can have the following meanings:

- verification complete;
- verification failed;
- incomplete verification.

6.3.4 Obtaining validation data

If the electronic signature appears to be conditionally valid (incomplete verification), the interface should suggest that the user makes a later attempt to capture the information to make it valid for the longer term.

6.3.5 User interface requirements

It is recommended to ensure user confidence in the electronic signature verification process by making the interface as easy to use as possible; by ensuring that the interface is accessible for all users, including people with special needs; and by reducing the probability of human error. In particular the dialogue system should:

- provide unambiguous user guidance based on relevant ergonomic standards/guidelines to cover how to use the signature verification system, and, if applicable, to install and configure the system.
- be self descriptive to the extent that each dialogue step is immediately understandable through feed back from the system or is explained to the verifier upon request.
- conform with usual verifier's expectations to the extent that it corresponds to the verifier's knowledge, education, experience and commonly accepted conventions.
- be adaptable to support the verifier's individual needs and preferences.
- be error tolerant if, despite evident errors in input, the intended result may be achieved with minimal corrective action.
- support informative error documents to lead the verifier forward.
- provide feedback to confirm that the action carried out by the verifier is correct (or incorrect).
- use standard conventions for the use of colours, e.g. red = error, green = go/proceed.
- be able, at any time, to cancel the current operation and return to the main menu; or, to exit the system completely.
- allow privacy for the individual, e.g. by making the information not accessible by others at the user interface.
- provide access to all people on equal terms (including first time users, children, the elderly). Physically handicapped and visually impaired people might require specific verification systems.

6.4 Machine verification

In contrast to the verification by human, the verification performed by a machine does not have a user interface. In this case signatures are automatically verified without direct user interaction. The result of the verification should be recorded (preferably in an audit trail) and may then be interpreted at a later time .

For automated processing, Application Programming Interfaces (APIs) may be used. Although there are many ways to construct such interfaces, they may be split into two categories:

a. APIs to extract the information contained in the electronic signature,

b. APIs to verify the electronic signature and obtain the validation data.

The first set of APIs allows extraction of the information contained in the electronic signature and obtaining the format of the electronic signature. It is then necessary to make sure that this format is supported by the APIs able to verify the electronic signature. The second set of APIs allows validation of and/or verification of the electronic signature and obtaining the signer information, the output status and the validation data. While the first set of APIs only offers suggestions about information contained in the electronic signature, the second set adds trust to this information.

Since there is no human being in front of the process, there is no need, in that case, for an interface to present the content of the SD.

Initial Verifications may require a third class of API: those needed to build the ES-C or ES-X formats.

6.5 Third-party verification

Verification may be sub-contracted to a third party. In such cases, the requester must be sure that the task has actually been performed by the chosen Third Party as requested. A data authentication service with integrity check must be used so that it can be made sure that the result comes from the right source. Since the Initial and Subsequent Verification processes are quite complex, this allows small clients (in the sense of a client/server model) to benefit from the use of electronic signatures without the need to directly support all the complexity of these processes. For “light” implementations, it is even possible to directly install a public signature verification key from a Third Party, i.e. without supporting a CA hierarchy.

7. Security Requirements for signature verification systems

7.1 Scope

Annex IV (b) of Dir.1999/93/EC [1] requires that "*the signature is reliably verified and the result of that verification is correctly displayed*" and (e) states: "*the result of verification and the signatory's identity are correctly displayed*".

In order to achieve this, reliable signature verification systems should have measures to counter attacks aiming to alter the verification process or to provide the verifier with false information on the process result.

These measures must be able to make the cost of a successful attack greater than its possible benefits.

This section gives guidance on implementing such systems.

All components of the signature verification system that interact with the Secure Signature Verification Process (see Section 5) should be implemented in a Secure Area.

Note: A Secure area is an area within which the storage and processing of data and the processes within this area are protected against successful tampering by means of special measures.

Regarding the degree of protection that can technically be achieved and the risk management applied by the verifier, the following three different levels of implementation should be considered:

- In a **Software Module** the security measures are implemented in software. The security that can be achieved in this way depends on the security of the operating environment. Especially for a PC with a standard operating system, adequate protection of data and processes by means of security measures implemented in software is a controversial topic among experts.
- In a **Tamper-evident module** security measures are implemented in such a way that, although manipulation cannot be prevented, the user can detect it. This means that a user can be prevented from unwittingly using a component of which the secure area has been manipulated. For a PC with a standard operating system, a tamper-evident module is only technologically possible at present using additional hardware.
- In a **Tamper-resistant module** security measures have been implemented in such a way that manipulation is not possible without unreasonable effort. The effort required for manipulation should be considered against the benefit derived from such manipulation. Such modules are presently only possible by using special hardware. A **Security module** is a component that as a whole constitutes a secure area created by means of a tamper-resistant module.

A mix of these different modules may be used to build a signature verification system. It should be noted that the overall security of the system is equal to the lowest security level of any of these modules.

7.2 Requirements for tamper-evident and tamper-resistant modules

Defining detailed requirements for hardware and software verification products is out of scope of the present document, therefore it is only provided guidance as to the security characteristics of products that would be used in the verification of electronic signatures, be they either (preferably) certified under a suitable formal evaluation or other form of assessment or (as a second choice) covered by a manufacturer's declaration of conformity to these requirements.

From the profile/security target it shall be possible to clearly identify which requirements are met and which are not met.

1. At the time of installation the integrity and authenticity of hardware and software should be ensured for all components.

2. Security-relevant data and processes in secure areas should be protected against unauthorised modification.

3. Unauthorised modifications of secure area hardware components should be recognisable.

7.3 Installation and verification assumptions

The components of the signature verification system may be implemented using a combination of trusted devices and other hardware and software.

The other hardware and software components may present various levels of security. In any case the security should first of all rely on a form of physical security. This may be obtained by placing the equipment in a presumably safe area, e.g. a home or a office. Alternatively, access security may be used to ensure that, after installation, what has been installed has not been modified.

Software providers of signature verification systems should provide the tools to perform checks to verify that the software is still unchanged at any time after installation.

Once a verification system has been set up, the user must be able to detect any manipulation before performing the signature-verification process, therefore security-relevant changes in components must be apparent for the user. Users must be able to verify the authenticity of these components before verifying a signature, in order to prevent "duping" with false data via manipulated technical components.

7.4 Requirements

The signature verification system should meet stated security targets to provide assurance for the whole system. The system consists on one side of the secure verification process (with its inputs and outputs) and on the other side of several interfaces.

The interfaces (whether human interfaces or programming interfaces) of an initial or subsequent signature verification system are used to present various information, including the output status to the signer.

The additional interfaces (whether human interfaces or programming interfaces) for a verification system are only used to provide for enhanced electronic signatures.

In the case of machine verification, the selection of the electronic signature to be verified is done by an application, while the presentation of the SD is only done in the case of the presence of and upon request by a individual.

The specific requirements to be met by a component are the same, but different components implementations provide different levels of confidence, depending on the environment where these components are used. Generally the entire system should be secure, which means that any information that is obtained from the whole system should be accurate.

7.4.1 Verification process

Where a signature policy is used the verification process should verify in a demonstrable way the electronic signature according to the rules, and in order to meet the objectives, of the relevant signature validation policy.

7.4.2 Selection of electronic signature for verification

The interface should allow the verifier to select the SD to be verified and, where applicable, the electronic signature to be verified.

7.4.3 Presentation of applicable Signature Policy

Where applicable, the interface should present to the verifier in a reliable, unambiguous and non-manipulated manner the identifier of the signature policy, a description of the expected application of this policy and the conditions that apply to the electronic signature.

7.4.4 Presentation of SD

The interface SHALL present to the verifier in a reliable, unambiguous and non-manipulated manner an unambiguous SD, i.e. the content of the SD to an adequate extent. The SD may be text, voice and video, which may be viewed, printed or listened to.

As stated in section 6.3.2, coded, or macro, instructions may be imbedded in the document in a way that the verifier is presented something different from what was signed. Therefore the user interface should issue a warning to the verifier if the document format may bear this kind of code. For this type of formats it is RECOMMENDED to use, where available, viewers which are able to notify the verifier when something has changed in the SD since it was signed and, possibly, which are the changes .

7.4.5 Presentation of signer information and output status

The interface should present to the verifier in a reliable, unambiguous and non-manipulated manner the name of the signer. The name should be extracted from the information contained in the Distinguished Name of the signer's certificate, together with the name of the CA. Other information like signing time, signer location, commitment type and roles, when present should also be presented. The output status of the process should be presented.

When initial verification is performed, one of the following three statuses should be presented:

- verification complete,
- verification failed,
- incomplete verification.

When Subsequent Verification is being performed, one of the following two statuses should be presented:

- verification complete,
- verification failed.

It may also happen that during the initial verification the enhanced electronic signature could not have been entirely implemented. In this case a subsequent verification may incur also in the “incomplete verification”. The verifier should be presented with this verification status and should be able to add the necessary missing information.

7.4.6 Requesting enhanced electronic signatures

This interface is specific to the initial verification process. When there may be a need for a subsequent verification, the interface should allow the verifier to request capturing of information that would make the electronic signature valid for the longer term. In that case, the initial verification process should provide the corresponding output.

As specified in the previous section 7.4.5, there may be a need that, even in what apparently is a Subsequent Verification, additional validation data are to be added to the signature. The Subsequent Verification related interface should not prevent the verifier to complete an Enhanced Electronic Signature.

8. Archive system

An Archived Electronic Signature (ES-A) is a form of signature that consists of both the Signed document and an supplemental Electronic Signature with additional validation data, basically an additional time stamp token and revocation status information about the previous Time-Stamp token. This form is necessary if the hash function and the cryptographic algorithms that were used at the time of the signature are likely to be no longer secure at the time of a later verification. In that case the hash function used by the Time Stamping Authority will have to withstand the further weakening possible to occur in the foreseeable future.

An archive system is composed of:

- an interface to enter the signed document and select the electronic signature(s) to be archived (there may be more than one electronic signature attached with the user data) – mandatory;
- an interface to obtain a trusted list of weak algorithms and key lengths as well as good algorithms and key lengths – conditionally present if the system does not already have this information hard coded;
- a user interface to inform the user if the electronic signature needs to be updated and to prompt him/her with a request for updating decision – recommended;
- a network interface to fetch information produced by CSPs, i.e. Time Stamp Token from a Time Stamping Authority, when it becomes necessary to extend the life-time of the cryptography – mandatory;
- a network interface to fetch the revocation status of the TSUs.

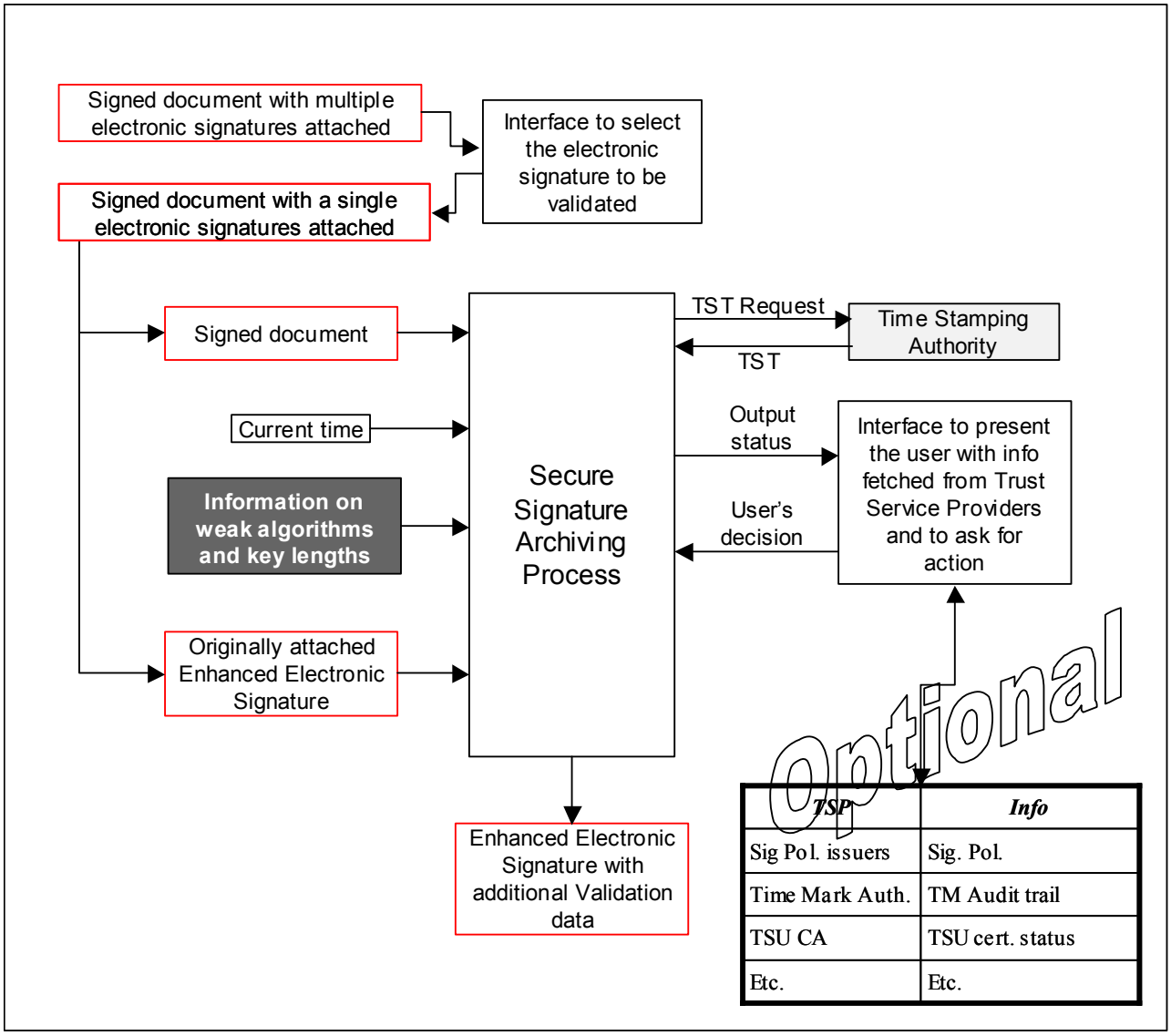


Figure 3

The process may need to be performed and iterated before the cryptographic algorithms used for generating the previous time stamp are no longer secure. An Archived Electronic Signature may thus bear one or more time stamps.

In any case it should be noted that the above provides trust only if the ES-A is built and all its versions are kept either by the verifier or by a trust service provider (e.g. a notary).

It is also necessary to add a new TST and then capture a CRL related to the previous TSU certificate in order to demonstrate that the TSU certificate was not revoked at the time the new time-stamp token was added. When later making a new Subsequent Verification, the revocation status of the last TST always has to be captured at the time of that verification, as specified in section 5.2.

Note: If the verifier has some other means that provide trust in the TSU certificate, then CRLs for TSUs do not have to be used.

Annex A - Annex IV from Dir.1999/93/EC

Recommendations for secure signature verification

During the signature-verification process it should be ensured with reasonable certainty that:

(a) the data used for verifying the signature correspond to the data displayed to the verifier;

(b) the signature is reliably verified and the result of that verification is correctly displayed;

(c) the verifier can, as necessary, reliably establish the contents of the signed data;

(d) the authenticity and validity of the certificate required at the time of signature verification are reliably verified;

(e) the result of verification and the signatory's identity are correctly displayed;

(f) the use of a pseudonym is clearly indicated; and

(g) any security-relevant changes can be detected.

Annex B Multiple Signatures

Some electronic documents may only be valid if they bear more than one electronic signature. This is the case generally when a contract is signed between two parties. The ordering of the signatures may or may not be important, i.e. one may or may not need to be applied before the other.

Several forms of multiple and counter signatures may need to be supported. They fall into the following categories:

- independent or parallel signatures;
- embedded or wrapping signatures;
- overall signatures.

Independent signatures are parallel signatures where the ordering of the signatures is not important. The capability to support and verify more than one independent signature over the same data should be provided. An electronic document with more than one independent signature can be considered as a collection of signed documents with an identical primary electronic document and therefore the same hash value (for any given algorithm which is applied). Whether or not the other signature attributes must coincide more or less depends on the signature context.

Independent signatures (i.e. parallel signatures) may be selected by using the interface able to select the electronic signature to be verified.

Embedded signatures are applied one after the other and are used where the order the signatures are applied is important. The first signature is called embedded and the second is the wrapping signature. The electronic document for the latter is the digital signature of the embedded signature only. The wrapping signature is therefore an unsigned attribute for the embedded signature but is expected to be verifiable by its own.

The embedded signature from the lowest level must be selectable, then wrapping signatures when they are present should also be selectable and verifiable independently.

Overall signatures differ from embedded signatures because the input for the signature is not only a previous signature but also the document itself. Overall signatures are used when the cryptography becomes weak and for archiving purposes and when they are present should also be selectable and verifiable independently.

Annex C - Time Stamping

Informative

In order to perform the validation, the certificate used by the signer at the time of the signature must be obtained, and its **validity at the time of the signature proven**. In this event, evidence must be provided that the document was signed when the certificate was valid, i.e. before the end of its validity (as indicated in the certificate validity period) and before it was revoked. Time-stamping can provide such evidence.

A time stamp by itself does not confirm the exact time when an electronic document was signed. A time stamp is obtained by sending the hash value of the given data to the TSA. The returned time-stamp is a signed document which contains the hash value, the identity of the TSA, and the time of stamping. This proves that the given data existed *before* the time of stamping.

If the hash of a digital signature is sent to a TSA and is time-stamped before the revocation of the certificate used to generate that signature, evidence will be provided that the digital signature was formed before the revocation of the public key certificate.

If a recipient wants to hold a valid electronic signature he will have to ensure that he has obtained a valid time stamp for it, before the certificate of the signer (and any certificate involved in the initial verification) is revoked. The sooner after the signature time, the better.

There will be some delay between the time a signature is created and the time the signer's digital signature is time-stamped. However, the longer this elapsed period the greater the risk of the signature being invalidated due to compromise or deliberate revocation of its corresponding certificate by the signer. Thus the signature policy should specify a maximum acceptable delay between the signing time as claimed by the signer and the time included within the timestamp.

In case of a compromise of the signer's key, the date of revocation of the signer's certificate can be compared with the date of the time stamp. If the date of the time stamp is earlier than the date of revocation, then it proves that the signature was applied before the revocation of the certificate hence the signature is valid. It also serves another purpose. Since a CA is no more responsible of taking care of the revocation of the certificates after the end of the validity period of the certificate, it is necessary to time stamp the signature to prove that is was performed before the end of the validity period of the certificate and therefore that it is based on a non-revoked certificate.

The exact signature time cannot be securely established. In fact the attempt to include within the signed data the purported time of the signature, as claimed by the signer, has only limited practical value, since such time has no intrinsic strength. Where necessary, it can be identified, at most, the time interval when the signature occurred. To this avail it may be useful to include a time stamp *inside* the signature. The drawback is the need to get access to a Time Stamping Authority before signing. However, this proves that the signature has been made *after* that date. Since the external time stamp indicates that the signature was made before that other date, this allows knowing that the signature was performed *between* these two dates.

Annex D - Signature policy and signature validation policy

Informative

D.1 The usefulness of a Signature policy

If verifiers (including an arbitrator) have to evaluate an electronic signature and to come to the same result, it is important that they use the same rules for verification. Such a set of rules is called a “signature policy”.

When signers sign a document, they may:

1. explicitly specify which signature policy is to be used by implementing the EPES format;
2. implicitly reference a signature policy within the signed document, or
3. implicitly reference a signature policy through other mechanisms or agreements.

In either case:

1. verifiers should use the signature policy which is explicitly specified or implied within the document, otherwise they might get a different result;
2. a clear definition of the signature policies is to be available to the verifying parties. Signature Policy Issuers need to make available the signature policies to the interested parties in a secure manner.

Where applicable, verifiers should also be able to figure out which signature policy was intended to be used. Thus signers should made sure that no ambiguity is ever possible on the selected signature policy that, if used, can be indicated implicitly or explicitly.

- It is implicit if, from the semantics of the SD or the type of data content being signed, it appears that some *other* documents, like national laws or private contractual agreements, specify that the signature must abide by specific rules, that, indeed, make up a signature policy. In this case, a given legal/contractual context may recognize a particular signature policy as meeting its requirements. However, automatic verification outside this context is not possible since, computers cannot determine only from the semantics of the signed data which signature policy shall be used. When the signature policy is implicitly referenced, verification may either automatically take place within a closed context (e.g. where only one type of document is being processed and where the semantics of the document is checked against that type) or may take place with human assistance when it can be proven, using out-of-bands means, which signature policy shall be used with this type of document.

- It is explicit, if an explicit reference to the signature policy is indicated by the signer within the electronic signature (and thus protected by the digital signature from the signer). In this case, the benefit is to allow processing the electronic signatures, even long after they have been generated and outside their original context of use (e.g. in front of a judge). When the signature policy is explicitly referenced, verification may automatically be performed without any human intelligence/assistance, if the Signature Policy is identifiable by a unique identifier, e.g. an OID (Object Identifier), and verifiable using a hash of the signature policy. In such a case, an electronic signature would include the unique identifier of the signature policy and the hash value of the signature policy and might also include a location (e.g. a URL) where a copy of the Signature Policy may be obtained.

The form and encoding of the specification of the signature policy is not mandated. However, for a given machine processable signature policy, there shall be one and only one definitive form. A signature policy shall be sufficiently definitive to avoid any ambiguity as to its implementation requirements. It SHOULD be absolutely clear under which conditions an electronic signature should be accepted.

A *signature policy* may be issued, for example, by a party relying on the electronic signatures and selected by the signer for use with that relying party. Alternatively, a signature policy may be established through an electronic trading association for use amongst its members. Both the signer and verifier use the same signature policy.

D.2 The publication of the Signature Policy

If a signature policy is used a signer, before signing, should be sure which security policy to apply. In the same way, when verifying an electronic signature, a verifier needs to make sure to use the right security policy. The security service able to provide that assurance is a combination of two basic security services: data origin authentication service and an integrity service. There are various ways to fulfil these two services in combination and this document does not mandate a particular way as long as the service is offered. However, it is recommended to support one of the three following ways that allow to make sure that the signature policy is genuine.

D.2.1 Using a trusted channel

Signature Policy Issuers may make available their policies either by placing them on a secure web site (e.g. accessed by using TLS / SSL) or by signing them. This is applicable for signers to generate the electronic signature and verifiers to verify the electronic signature soon after it has been generated, but may not be practical for long term verification, because it may not be easy to verify the digital signature of the issuer in the long term, hence other methods are necessary.

D.2.3 Using trusted Repositories of registered security policies

Signature Policy Issuers may disappear and situations where they cannot provide anymore a trusted copy of the policies they have issued may appear. It thus becomes necessary to rely on a third party. For long term verification of Electronic Signatures the use of trusted Repositories of registered Signature Policies is thus needed. In that case, it becomes sufficient to connect to these Repositories and access them using a data origin authentication service combined with an integrity service. This may be achieved using TLS (Transport Layer Security). Signature Policies must be kept in Trusted Repositories as long as there is a need to verify an electronic signature that has been made against these Signature Policies.

D.2.3 Using a trusted media

Each Signature Policy Issuer may provide a trusted media to the users, e.g. in the form of a CD-ROM, that can be authenticated as issued by the Signature Policy Issuer.

D.3 The main contents of the Signature Policy

A signature policy MUST include:

- a unique identifier or unambiguous identification of the signature policy;
- the signature policy issuer name;
- the issuing date of the signature policy;
- the **field of application** of the signature policy;
- a **signature validation policy**.

D.3.1 Field of application

The **field of application** of the signature policy is a description of the expected application of this policy and the conditions that apply to the electronic signature. This part of the signature policy can be assessed to meet the requirements of the legal and contractual context in which it is being applied. It is intended to be available for display (or listening) both by the signer or the verifier. This part of the signature policy is similar to the contractual terms that may be found on the back of a contract that apply to all the commitments made under that signature policy.

D.3.2 Signature Validation Policy

The **signature validation policy** specifies the technical rules to be followed by the signer and the verifiers to process the electronic signature. These rules allow for the initial and usual verifications of electronic signatures issued under that form of signature policy. They may be either described as free text or be described in a machine processable language.

The Signature Validation Policy includes rules defining the components of the electronic signature that should be provided by the signer with data required by the verifier to provide long term proof, as well as rules regarding use of TSPs (CA, OCSP servers, Attribute Authorities, Time Stamping Authorities). For that reason it should include:

- rules for Certification path construction/verification;
- rules for use of Revocation Status Information (e.g. CRLs or OCSP responses);
- rules for use of Timing information, Time-Marking and/or Time Stamping;
- signature validation data to be provided by the signer;
- signature validation data to be collected by verifier.

D.4 Categories of verification systems

Verification of an electronic signature against a signature policy requires a signature verification system able to process the signature policy. There may be some cases where specific parameters from the security policy (e.g. root keys) are downloaded into the system. In such a case the source of the download should be authenticated and the integrity of the downloaded data should be verified. There are basically two categories of verification systems.

D.4.1 Specific signature policies

These systems only support **specific signature policies**. The verification process implemented by the system should conform to a human readable description provided all the processing rules of the signature policy are clearly defined. However, if additional policies need to be supported, then such an implementation would generally need to be customized for each additional policy. This type of implementation may be simpler to implement initially, but can be difficult to enhance to support numerous additional signature policies.

Each verification system must be evaluated against each security policy that is claimed to be supported by the system.

D.4.2 Dynamically programmable signature policies

These systems are **dynamically programmable** and able to adapt their rules in accordance with a description of the signature policy provided in a computer-processable language (e.g. ASN.1). This type of implementation is able to support multiple signature policies without being modified every time, provided all the verification rules specified as part of the signature policy are known by the implementation (i.e. only requires modification if there are additional rules specified).

Verification systems must be evaluated once against the security policy descriptive language.

It should also include:

- the period during which signatures can be performed under that policy,
- a list of recognized commitment types;
- rules for the use of signer roles;
- any constraints on signature algorithms and key lengths;

- other signature policy rules required to meet the objectives of the signature.

The signature validation policy may recognize one or more types of commitment as being supported by electronic signatures produced under the security policy.

If an electronic signature does not contain a recognized commitment type then the semantics of the electronic signature is dependent on the data being signed and the context in which it is being used.

Annex E – Examples of user environments

Informative

Four main environments have been considered: the home environment, the office environment, the public environment and the mobile environment. The ways to meet an acceptable assurance level are not identical. Basically security is guaranteed by a combination of technical measures and of procedural or organizational measures. For each of these environments these two components exist but are not equally weighted.

The classification of the different environments is strongly related to the difference in the responsibility/liability for the various environments.

E.1 Home environment

In a private application environment, the user is provided with trust in his private equipment, because the equipment is either placed under its exclusive control or placed under the control of a small group of people who all trust each other. He may use various kinds of equipment, e.g.:

- a Desktop (PC),
- an interactive television (smart TV)
- a Network Computer,
- a Personal Digital Assistant (PDA),
- an Internet Phone.

The software able to verify electronic signatures will use and co-exist with other software (the operating system and the applications). It must be made absolutely sure that such other software is not malicious or cannot interfere with this software.

The software that verifies electronic signatures may be either pre-loaded by the user or downloaded at the time it is needed. In both cases, the user must rely on the source where from he obtained the software. However, the confidence is not obtained in the same way.

If pre-loaded, the confidence will comes both from the shopping place and the form of the package that contains the software. The package(s) should have a clear indication of the characteristics that are supported, in particular:

- which electronic signature formats are supported,
- which signature policies or which signature policy formats, are supported, where applicable
- which document formats, may be presented (WIPIWIS).

Note: when signature policies are used the two first characteristics may be combined, while the third one should be kept separate in order to be able to be upgraded to support new forms of documents.

The packages should also have a clear marking that they fulfil the requirements laid down in the present document. This could be achieved by using a logo.

Then the user needs to install the software. In order to increase the level of security, the equipment could, in addition, contain a public key used to verify that the software being loaded is correctly signed.

If downloaded at the time of verification, the confidence comes from the fact that the software is signed and that the latest version is being downloaded. The security of the downloading process relies on a public key that is provided with the equipment or that is installed once by the user.

Since the user directly participates in the installation, he is directly responsible to verify that the correct procedures have been followed.

Once the installation has been performed, the user must make sure that no one can modify the installed software, without this being detected. This may be achieved by using only physical security (e.g. a door lock) or a combination of physical security and software (e.g. boot-protection, anti-virus, hard disk encryption).

E.2 Office environment

In an office environment, the organisation the user belongs to has the responsibility to provide the adequate tools for day to day working. The user may use various kinds of equipment, e.g.:

- a Desktop (PC),
- a Laptop,
- a Personal Digital Assistant (PDA),
- a Mobile Phone.

The software able to support the functionality may be either pre-loaded by the people in charge of the user's equipment or downloaded at the time it is needed.

In the first case, the organization must rely on the source where from it obtained the software. The confidence will come both from the shopping place and the packages that contains the software. The package(s) should have a clear indication of the characteristics that are supported, in particular:

- which electronic signature formats, are supported,
- which signature policies or which signature policy formats, are supported, where applicable,
- which document formats may be presented (WIPIWIS).

The packages should also have a clear marking that they fulfil the requirements laid down in the present document. This could be achieved by using a logo.

Then the people in charge of the user's equipment need to install the software. This can be done locally on each equipment by downloading the software from a trusted server from the organization or be done remotely.

In the second case, the user must rely on the source of code that is provided externally. That code must be signed and it must be made sure that the latest version is being downloaded. The security of the downloading process relies on a public key that is provided with the equipment or that is installed once by the people in charge of the user's equipment.

Since the user does not directly participate in the installation, he is not directly responsible to verify that the correct procedures have been followed. However, he/she might have the possibility to modify the software, or its configuration, afterwards. This capability depends from the operating system being used, in particular whether it makes a difference between administrators and users.

Once the installation has been performed, the organization must make sure that no one can modify the installed software without this being detected. This may be achieved by using a combination of physical security and software (e.g. boot-protection, anti-virus, hard disk encryption).

E.3 Public environment

In a public environment, the organisation to which the verification system belongs to has the responsibility to provide a trustworthy system. The user may use various kinds of equipment, e.g.:

- a retail Point-of-Sale (POS),
- an Automated Teller Machine (ATM),
- a public service point.

The installation phase must be carried out under the responsibility of the organisation owning the system, which means that the name and address of that organization must be clearly identified on the system itself. Means to reach that organization by phone, surface mail, E-mail or web should be indicated. The organization must then make sure that no one can modify the installation without placing the system out of service.

The user will get confidence in such a system not only because it will look like a “real” system which bears a clear marking (e.g. logo) of what it is intended for, but also because it is placed in an environment where it would be difficult, without being noticed, to install a fake system.

In order to educate users with the view of “real” systems, users should be advertised personally or/and through the press of such an appearance.

E.4 Mobile environment

In a mobile environment, some equipment that may be used as a stand alone device and are already considered in an office or home environment may be used, e.g. a Laptop connected to a telephone line. Although the delimitation between stand alone systems and systems that only work connected is hard to denote (e.g. a Personal Digital Assistant (PDA) connected or not to a phone line), some devices are meant to be used in a connected environment, e.g. a mobile phone. In such a case, the user is provided with trust in his private equipment, first because the equipment is provided by a manufacturer which should provide some assurance about the properties of the device. Secondly, because the device then is placed under the exclusive control of the user, who is going to personalize the equipment with user specific data, e.g. stored in a SIM card.

The verification software able to support the functionality may either be put on the device by the manufacturer, be installed by the user or be downloaded at the time it is needed (e.g. using MExE). MExE has an architecture for verifying downloaded content which is itself based on digital signatures. The user therefore relies on the source of the software for its content, but can be given assurance as to the integrity and authenticity of the content by software that the terminal manufacturer has put on the device.

The downloaded software may be provided by the ’phone company and the security of the loading may then come directly from the protocol being used between the mobile equipment and the network. Alternatively, this software may be provided by a content provider and the security of the loading may then come from the security of the proxy server.

In order to authenticate the software some root keys will have to be used. They may be manually installed by the user. This may be done by fetching the information from some place and then verifying that the hash of what has been downloaded matches a hash value provided by out-of-bands means. They may also be put on the device by the manufacturer or on the SIM or WIM. User download is what could be implemented.

The software that has been installed or downloaded shall then have the properties to authenticate the source of the signature policies.

Document History

TO BE REMOVED BEFORE PUBLISHING

Date	Version	Notes
July 2001	CWA 14171:2001 E	
May 2003	Draft CWA 14171 v 2.1.2	Main changes: 1. Signature Policies have been made no more mandatory. Their explanation has been shortened, and moved to an Annex along with a digest of their references throughout the document. 2. The “grace period” has been detailed. 3. The term “Enhanced Electronic Signature” has been inherited from the EESSI Final Report – July 1999 – to address all ES beyond the basic ES, previously called “Augmented ES”. 4. Reference list has been cleared of unreferenced items and generally reviewed. 5. A distinction of signatures has been done on the basis of their “expected lifetime”: Ephemeral, Short term, Long term. This has been done to better address in which case the single component is needed. 6. The term “Subsequent verification” has been used instead of “Usual verification” 7. More detailed explanations of some rationales have been inserted. 8. A clear specification has been added that there is no possibility to oblige signers and initial verifiers to implement Enhanced ESs, since no one can be forced to do something that can be used against himself (sounds too detective movie like?) 9. A reference to the now issued new RFCs and ETSI TS and CWAs has been added. 10. RFC 3280 has an extensive explanation and description of the certification path verification algorithm, so the CWA 14171 related section has been replaced with a reference to it. 11. A reference to the now under way ETSI STF 220 Task 1 output has been added. 12. Figures have been updated accordingly. 13. Examples of different verification environments have been moved to an Annex. 14. Requirements section for HW tamper resistant modules has been slimmed down. 15. Conformity assessment section has been slimmed

		down a lot, since CWA 14172 part 4 defines the relevant criteria. 16. Legal parts have been removed, since the legal subject is now covered extensively in CWA 14365. 17. Similarly the Annex on “How may a verifier really know who the signer is ?” has been removed, since it addresses methods which are outside the specific verification process. 18. Most SHALL/ MUST have been changed into SHOULD, because: a. this CWA is issued as a way to implement the Directive Annex IV, which addresses recommendations; b. nobody can be compelled to enact anything that can be used against himself c. the signature cannot be deprived of validity just because it lacks some technical feature d. the border between what must be done by the signer and what is up to the verifier is very blurred. 19. Annex F added.
August 2003	Draft CWA 14171 v 2.1.3	Draft document for formal voting
September 2003	Draft CWA 14171 v 2.1.4	Approved document for CEN publication
October 2003	Draft CWA 14171 v 2.1.5	Pre-publication editorial refinement
October 2003	Draft CWA 14171 v 2.1.6	Pre-publication editorial refinement
October 2003	Draft CWA 14171 v 2.2.1	Changes based on comments from D. Pinkas
October 2003	Draft CWA 14171 v 2.2.2	Pre-voting editorial refinement; draft document for formal voting
February 2004	Draft CWA 14171 v 2.2.3	Approved, corrected typo in section 5.2
March 2004	Draft CWA 14171 v 2.2.4	Approved, corrected typo in section 5.6.3; for CEN publishing

Bibliography

The following material provides supporting information.

- European Electronic Signature Standardization Initiative (EESSI) – Final Report of the EESSI Expert Team 20th July 1999
- ISO 7498-2:1989 Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture
- ISO/IEC 10118-1:1994 Information technology – Security techniques – Hash-functions – Part 1: General