

## Technical standard for the electronic filing of European patent applications and subsequent documents

### 1 Background

This document contains the technical standards for the electronic filing of documents with the EPO. It is based on the Trilateral Public Key Infrastructure (PKI)-based standard that has been incorporated into Annex F, Appendix I of the PCT Administrative Instructions.

A PKI environment provides a suite of services for processing sensitive information. Through the use of cryptography, PKI can satisfy the requirements for:

- (a) Authentication – by ensuring that transmissions, messages and originators are valid, and that a recipient is authorised to receive specific categories of information.
- (b) Data integrity – by ensuring that data is unchanged from its source and has not been accidentally or maliciously modified, altered, or destroyed.
- (c) Non-repudiation – by ensuring strong and substantial evidence is available to the sender of data that the data has been delivered (with the co-operation of the recipient), and to the recipient of the sender's identity, so that neither party can successfully deny having possessed the data, and a third party can verify its integrity and origin.
- (d) Confidentiality – by ensuring that the information can be read by authorised entities only.

This standard sets out the mandatory requirements for all parties participating in electronic filing, as well as a number of optional requirements.

### 2 Scope

This technical standard covers requirements in the following areas:

- (a) Security and PKI
- (b) Electronic signatures
- (c) Document format requirements
- (d) Submission

### 3 Security and PKI

#### 3.1 Public Key Infrastructure

In this standard, packaging and transmission are performed using PKI technology. When feasible alternative security technologies become available, they may be incorporated in updates to the standard.

PKI must be implemented in accordance with the recommendations established by the Internet Engineering Task Force (IETF) Working Group on PKI Interoperability (PKIX) and documented in IETF RFC 2459.

Separate key pairs and digital certificates must be used for the digital signature and encryption.

#### 3.2 Digital certificates

Where the standard specifies use of a digital certificate, the certificate must comply with the International Telecommunication Union (ITU) X.509 (version 3) recommendation for certificate format.

A digital certificate is required when communicating with the EPO online.

The standard provides for two classes of digital certificate:

*High-level certificate:* a digital certificate issued by a certification authority to the applicant, which can be used to authenticate the identity of the applicant. The certification authority must appear on the list of "recognised" certification authorities published by the EPO (see 3.3 below).

*Low-level certificate:* a digital certificate provided by the EPO to the applicant on request. To receive a low-level certificate, the applicant must provide his name and e-mail address, but is not required to furnish proof of identity.

#### 3.3 Certification authorities

The EPO will specify which certification authorities it accepts. This list of "recognised" certification authorities will include a link to the published PKI policy statement of each of these authorities.

Recognised certification authorities are responsible for maintaining the accuracy of the electronic certificates that "prove" a party is who he says he is. Certification authorities store certificate information for all the certificates they issue in a directory structure complying with ITU recommendation X.500. Such systems provide an external interface for publishing and retrieving user digital certificates that complies with the Lightweight Directory Access Protocol (LDAP) using the IETF Network Working Group's RFC 1777 dated March 1995. In addition, certification authorities publish revocation information about certificates drawn up in accordance with the X.509 standard.

The EPO will subscribe to this revocation information. Whenever a certificate is used to authenticate an individual, the EPO will consult the revocation information published by the certification authority concerned to ensure that the certificate has not been revoked.

#### 3.4 Digital signatures

Digital signatures used to sign electronic documents for electronic filing must conform to the format and practice specified in RSA Laboratories' PKCS#7 Cryptographic Message Syntax Standard (version 1.5) with regard to the definition of the signed-data content type.

To build these signatures, a certificate meeting the requirements set out in Section 3.2 above must be used.

All digital signatures must be encoded using the distinguished encoding rules (DER) defined in ITU recommendation X.690.

### 3.5 Cryptographic algorithms

Both symmetric (secret key) and asymmetric (public key) algorithms may be used as required. Algorithms prohibited under the national law of a country may not be used for the electronic filing of documents from that country. Algorithms implemented in hardware or software may not be used in any manner contrary to the export restrictions of the country of origin of the hardware or software.

Where possible, the rsaEncryption algorithm is to be used for asymmetric encryption and the des-EDE3-CBC algorithm for symmetric encryption. The same asymmetric encryption algorithm should be used to create digital certificates, digital signatures and envelopes.

### 3.6 Data enveloping

Electronic document data that is encrypted to ensure confidentiality for electronic filing must conform to the format and practice specified in RSA Laboratories' PKCS#7 Cryptographic Message Syntax Standard (version 1.5) with regard to the definition of the signed and enveloped data content type.

### 3.7 Message digest algorithms

The message stream must be input to the strong one-way message digest algorithm SHA-1 to create a message digest.

## 4 Signature mechanisms

This standard provides for a number of signature types acceptable for electronic filing:

- (a) Basic electronic signatures
  - (i) Facsimile image of the user's signature
  - (ii) Text string
- (b) Enhanced electronic signature
  - (i) PKCS#7 digital signature

**NOTE:** Although users may choose not to utilise an enhanced electronic signature mechanism for the document itself, a PKCS#7 digital signature is required to package the wrapped application document as described in section 5.3. See Section 6.1 for an example of a wrapped and signed package.

The basic electronic signature is encoded within the "party" structure of the XML document as specified by the portion of the Document Type Definition (DTD) shown below:

```

...
<!ELEMENT electronic-signature (basic-signature, enhanced-signature?) >
<!ATTLIST electronic-signature
  DATE-SIGNEDC DATA #REQUIRED
  PLACE-SIGNEDC DATA #IMPLIED >

  <!ELEMENT basic-signature (fax | text-string) >

    <!ELEMENT fax EMPTY >
    <!ATTLIST fax
      FILE-NAME ENTITY #REQUIRED >

    <!ELEMENT text-string (#PCDATA) >

  <!ELEMENT enhanced-signature (pkcs7) >
  <!ELEMENT pkcs7 EMPTY >
...

```

A basic electronic signature within an XML document may be supplemented by the addition of a digital signature to the wrapped application documents.

**4.1 Facsimile signature**

To create this type of signature, the XML file must include the <fax> element and an external entity reference set in the FILE-NAME attribute that points to the file containing the bitmap of the signature, as shown below:

```

...
<electronic-signature DATE-SIGNED="01/01/2000">
  <basic-signature>
    <fax FILE-NAME="signature.tif" />
  </basic-signature>
</electronic-signature>
...
    
```

This bitmap file must be a 300dpi single strip, Intel encoded TIFF Group 4 image or a JFIF (JPEG) file.

**4.2 Text string signature**

To create this type of signature, the XML document must include the <text-string> element containing a text string that represents the user's "wet" (ink) signature, enclosed in slash "/" characters, as shown in the example below:

```

...
<electronic-signature DATE-SIGNED="01/01/2000">
  <basic-signature>
    <text-string>/janedoe/</text-string>
  </basic-signature>
</electronic-signature>
...
    
```

The text string must be a string of characters, not including the forward slash "/" character, chosen by the user as his electronic signature, as shown in the following examples:

```

...
<text-string>/John Smith/</text-string>
<text-string>/Tobeornottobe/</text-string>
<text-string>/1345728625235/</text-string>
<text-string>/Günter François/</text-string>
...
    
```

**4.3 PKCS#7 digital signature**

The PKCS#7 signed data type is generated from the electronic message by the signer, who uses his private signing key to encrypt the message digest. It includes a copy of the digital certificate of the signer when sent.

The use of a PKCS#7 signature must be indicated in the XML file by the <pkcs7> element, as shown below:

```

...
<electronic-signature DATE-SIGNED="01/01/2000">
  <enhanced-signature>
    <pkcs7 />
  </enhanced-signature>
</electronic-signature>
...
    
```

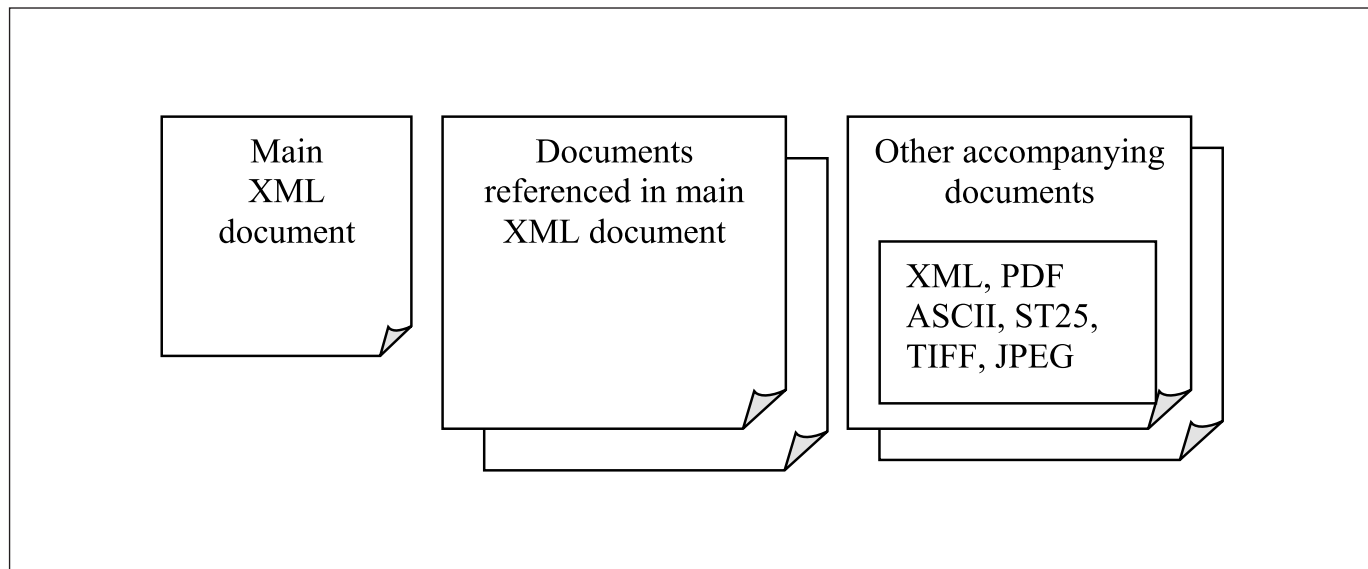
**5 Data format requirements**

The document packaging mechanism is used to combine the data about what is being transmitted with the contents of the transmission to form a single binary object called a wrapped application document (WAD), and then to apply the appropriate digital signatures and encryption.

**5.1 Document preparation**

For each document filed there is a main XML document that may explicitly reference all documents to be sent in a single package. These referenced documents are logically part of the main document (eg a new patent application). In addition, a filing may include other accompanying documents (eg designation of inventor or fee payment).

The main XML document must conform to one of the DTDs specified below. The referenced documents (external entities) are typically embedded images, tables, drawings or other compound documents and may be encoded as either XML, ST25, PDF, ASCII, TIFF or JFIF(JPEG). The accompanying documents are separate, but related, documents that may be encoded as either XML, ST25, PDF, ASCII or Image. Any accompanying XML documents must also conform to one of the DTDs specified below.



**5.1.1 Character-coded formats**

**5.1.1.1 XML**

All XML documents must conform to one of the DTDs specified below. Applicants will be able to create XML documents conforming to this standard by using the EPO's client software for electronic filing.

The coded character set used for all XML documents must be confined within that specified by ISO/IEC 10646:2000 (Unicode 3.0). The standard character-encoding scheme for XML documents is UTF-8.

**5.1.1.2 ST.25**

A document created using WIPO ST.25 SGML tags for sequence listings may be included in a WAD as an external document.

**5.1.1.3 ASCII**

A document created as plain ASCII text may be included in a WAD as an external document. In this case, the main XML document must include the code page of the ASCII text.

**5.1.2 PDF**

PDF documents for use in electronic filing must meet the following requirements:

- (a) PDF V1.3 compatible
- (b) Non-compressed text to facilitate searching
- (c) Unencrypted text
- (d) No digital signatures
- (e) No embedded OLE objects
- (f) All fonts must either be embedded, standard PS17 or built from Adobe® Multiple Master (MM) fonts

The PDF format has become the de facto standard for the exchange of formatted documents on the Internet.

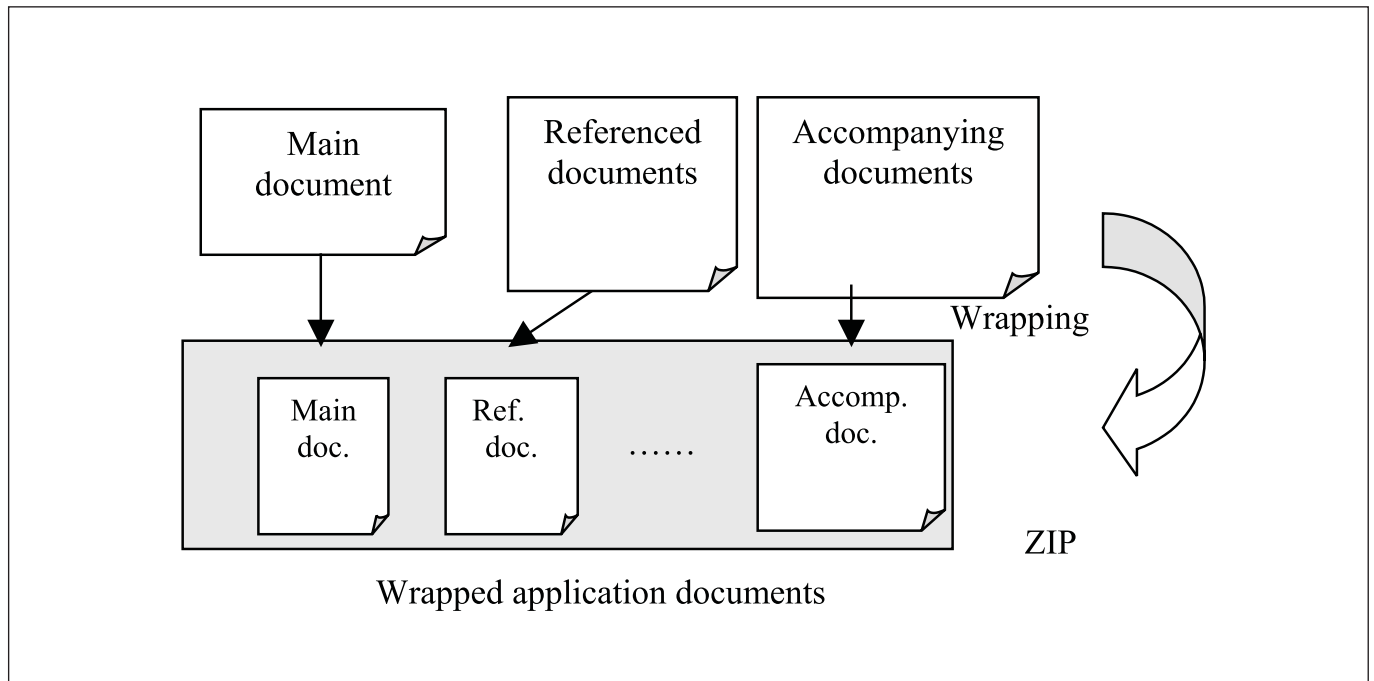
**5.1.3 Images**

Facsimile images used in electronic filing must meet the following requirements:

- Format
  - TIFF V6.0 with Group 4 compression, single strip, Intel encoded or
  - JFIF(JPEG)
- 200, 300 or 400 dpi
- A4 size

**5.2 Wrapping documents**

The main document and any externally referenced documents and accompanying documents are wrapped and treated as one data block. This data block, called the wrapped application documents (WAD), is created using the ZIP wrapping standard. Applicants must use ZIP format archiving and compression software to package the document files constituting an electronic application.



The software used to create the ZIP file must conform to the ZIP file format specification as published in the PKWARE® PKZIP® Application Note (revised: 8.1.1998).

The files to be zipped must include all parts of the document identified elsewhere in this standard. All external files referenced by the application must be included in the ZIP file submission. File names included in the central directory of the ZIP file must comply with the specification for the applicable operating system.

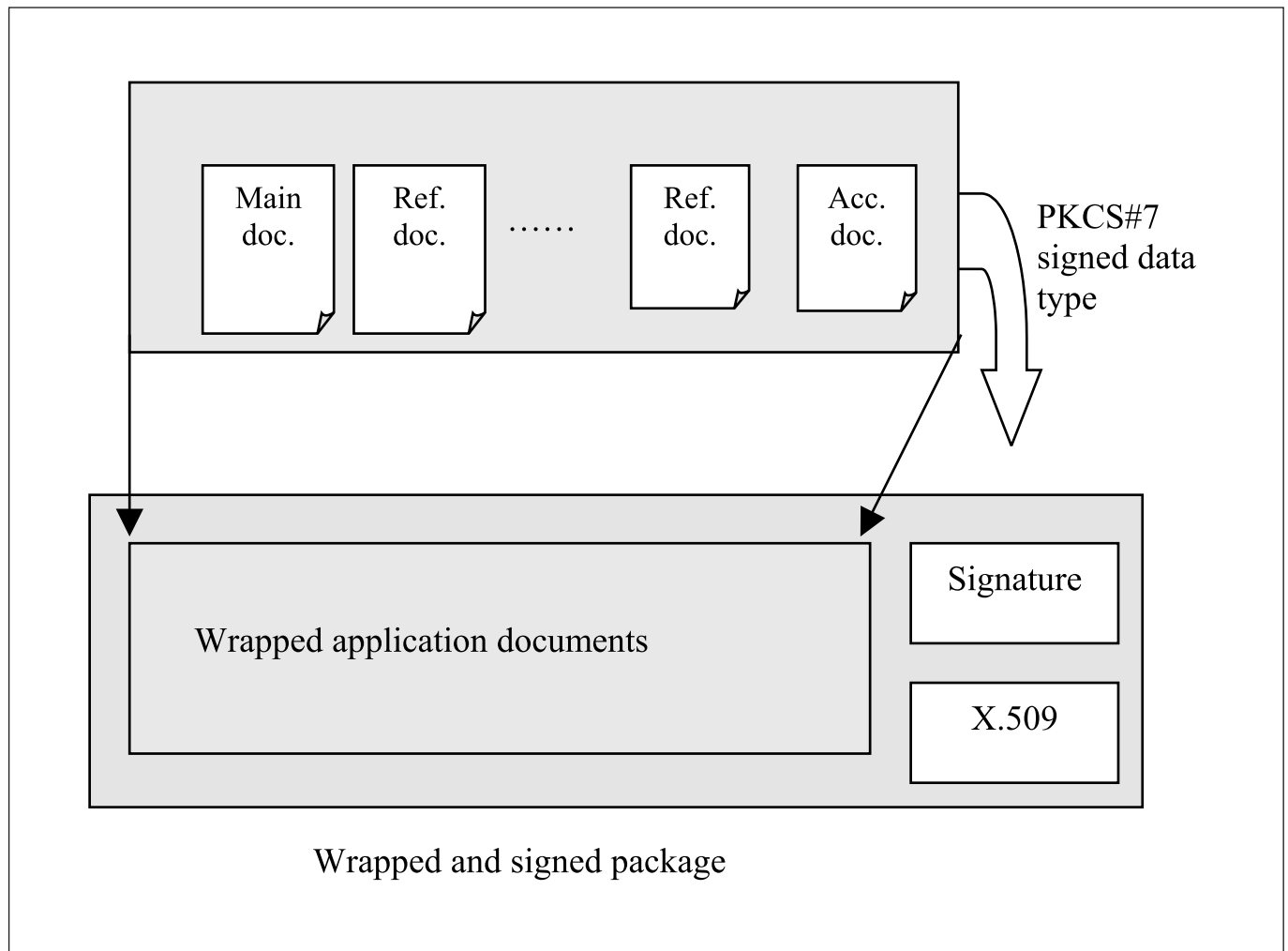
All ZIP files must have a flat directory structure. If a collection of files needs to be embedded in the ZIP file, then these should be included as a single flat embedded ZIP file.

The ZIP standard allows the compression software to select from among a number of compression algorithms. The default compression method must be "Deflation".

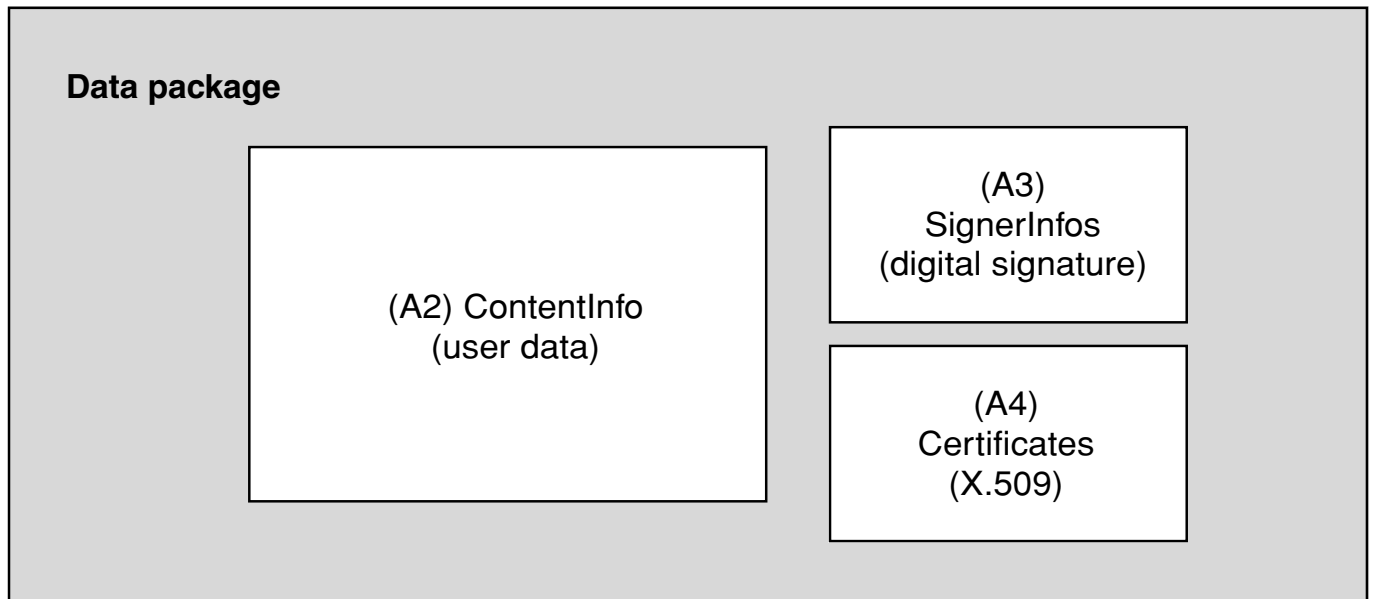
**5.3 Signing wrapped application documents**

To bind the person creating the package to the electronic wrapped application documents, a digital signature is added to create the wrapped and signed package. The purpose of adding the signature is to identify the person creating the package and to enable the recipient to detect any unauthorised alteration during transmission.

PKCS#7 is used to produce a signed data type for the signature.



**(A1) SignedData <top level>**  
**(PKCS#7 digital envelope for signature)**



Rules for producing the PKCS#7 digital envelope for certification

Object identifier for sha-1	The object identifier adopted for SHA-1 is defined in OIW interconnection protocols (Part 12) as follows: <b>Sha-1 OBJECT IDENTIFIER ::= {iso (1) identified-organization(3) oiw(14) secsig(3) algorithm(2) 26}</b>
Object identifier for RSA encryption	The object identifier for RSA encryption is defined in <i>RSA Encryption Standard PKCS#1</i> as follows: <b>Pkcs-1 OBJECT IDENTIFIER ::= iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1) 1}</b> <b>RsaEncryption OBJECT IDENTIFIER ::= {pkcs-1 1}</b>
Object identifier for triple DES	<b>dES-EDE3-CBC OBJECT IDENTIFIER ::= {iso(1) member-body(2) us(840) rsadsi(113549) encryptionAlgorithm(3) 7}</b>

**Table A1 SignedData – top level**

No.	Item name	PKCS#7 item	Content
1	Version	Version	Set integer value '1'
2	Set of algorithm identifiers	DigestAlgorithms	
2.1	Algorithm identifier	AlgorithmIdentifier	Set ONE <b>set</b> of algorithm identifiers {sha-1} only
3	Content information	ContentInfo	Set one content information (see table A2)
4	Certificates	Certificates	Set one Certificates (see table A4)
5	Certificate revocation lists	Crls	Not used (set no data)
6	Signer information	SignerInfos	Set one SignerInfos (see table A3)

**Table A2 ContentInfo** – top level

No.	Item name	PKCS#7 item	Content
1	Content type	ContentType	Set object identifier {pkcs-7 1}
2	Content	Content	Set user data (binary)

**Table A3 SignerInfos** – top level

No.	Item name	PKCS#7 item	Content
1	Version	Version	Set integer value '1'
2	Issuer and serial number	IssuerAndSerialNumber	Issuer of certificate and certificate serial number in acc. with X.509 (signer's certificate)
3	Set of digest algorithms	DigestAlgorithm	
3.1	Algorithm identifier	AlgorithmIdentifier	Set ONE <b>set</b> of algorithm identifiers {sha-1} only to make digest of digital signature
4	Authenticated attributes	AuthenticatedAttributes	Not used (set no data)
5	Digest encryption algorithm	DigestEncryptionAlgorithm	Algorithm OBJECT identifier of digest encryption (recommended algorithm: rsaEncryption)
6	Encrypted digest	EncryptedDigest	Digest data encrypted using signer's private key
7	Unauthenticated attributes	UnauthenticatedAttributes	Not used (set no data)

**Table A4 Certificates** – top level

No.	Item name	PKCS#7 item	Content
1	Set of certificates	ExtendedCertificatesAndCertificates	
1.1	X.509 certificate	Certificate (defined in X.509)	Set ONE <b>set</b> of X.509 certificate data only

**6 Submission**

**6.1 Transmission package**

The EPO may decide not to use the enveloping mechanism described in this section as the encryption mechanism for transmission where channel level encryption such as SSL or physical media such as CD-R are used.

The actual transmission data exchanged between the applicant and the EPO is called a package.

A package contains various data items depending on the type of package. These include:

1. Header object data item
2. Wrapped and signed package made by wrapping and signing the application documents
3. Transmission data such as time of transmission.

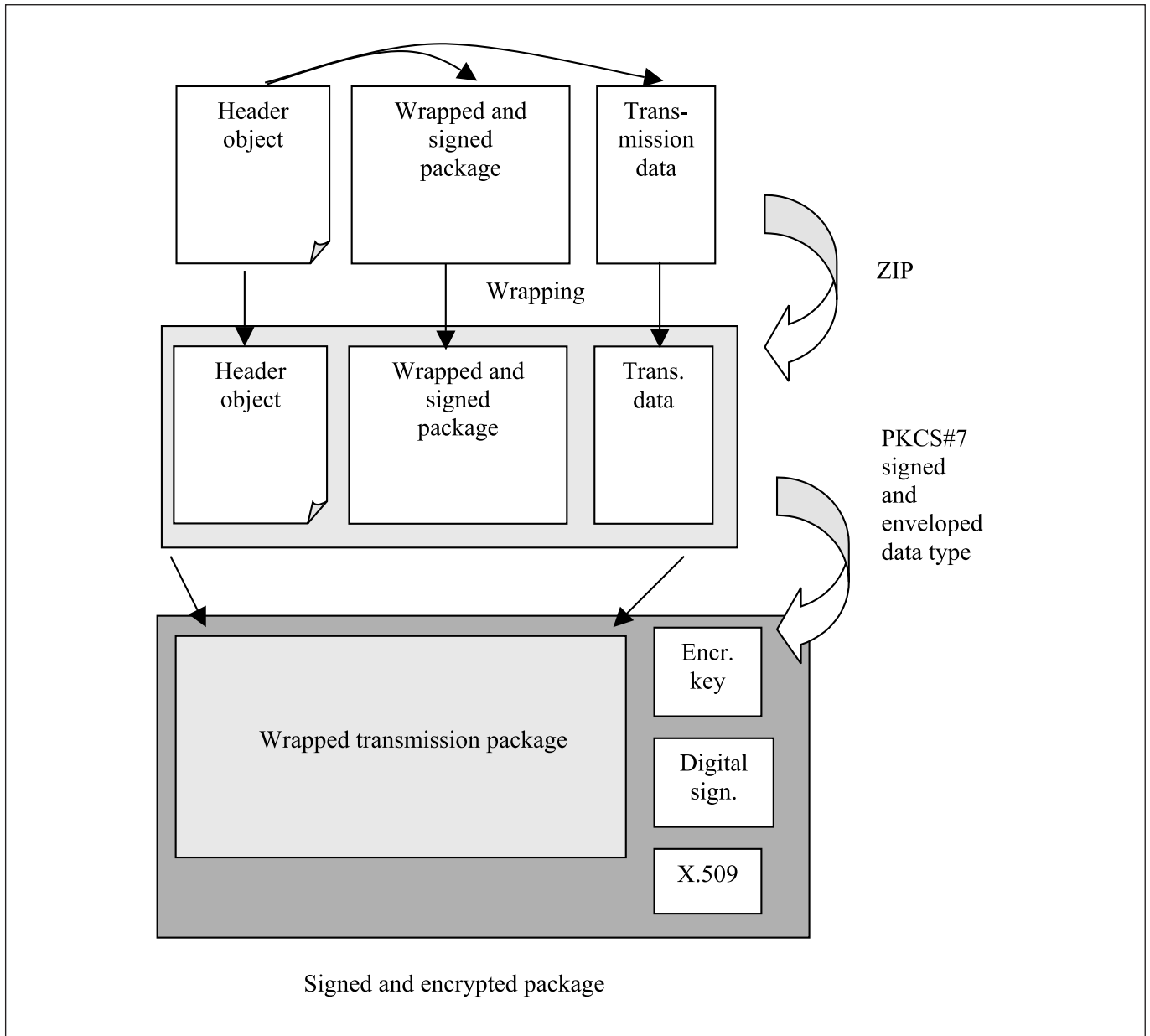
The header object data item indicates the package type, file name of data item, etc. It is always found in the signed and encrypted package, and is written in XML.

The procedure for creating signed and encrypted packages is as follows:

- (a) Create a wrapped transmission package by wrapping the wrapped and signed package with the data items used for transmission using ZIP
- (b) Create a signed and encrypted package for network transmission by encrypting using the PKCS#7 signed and enveloped data type.

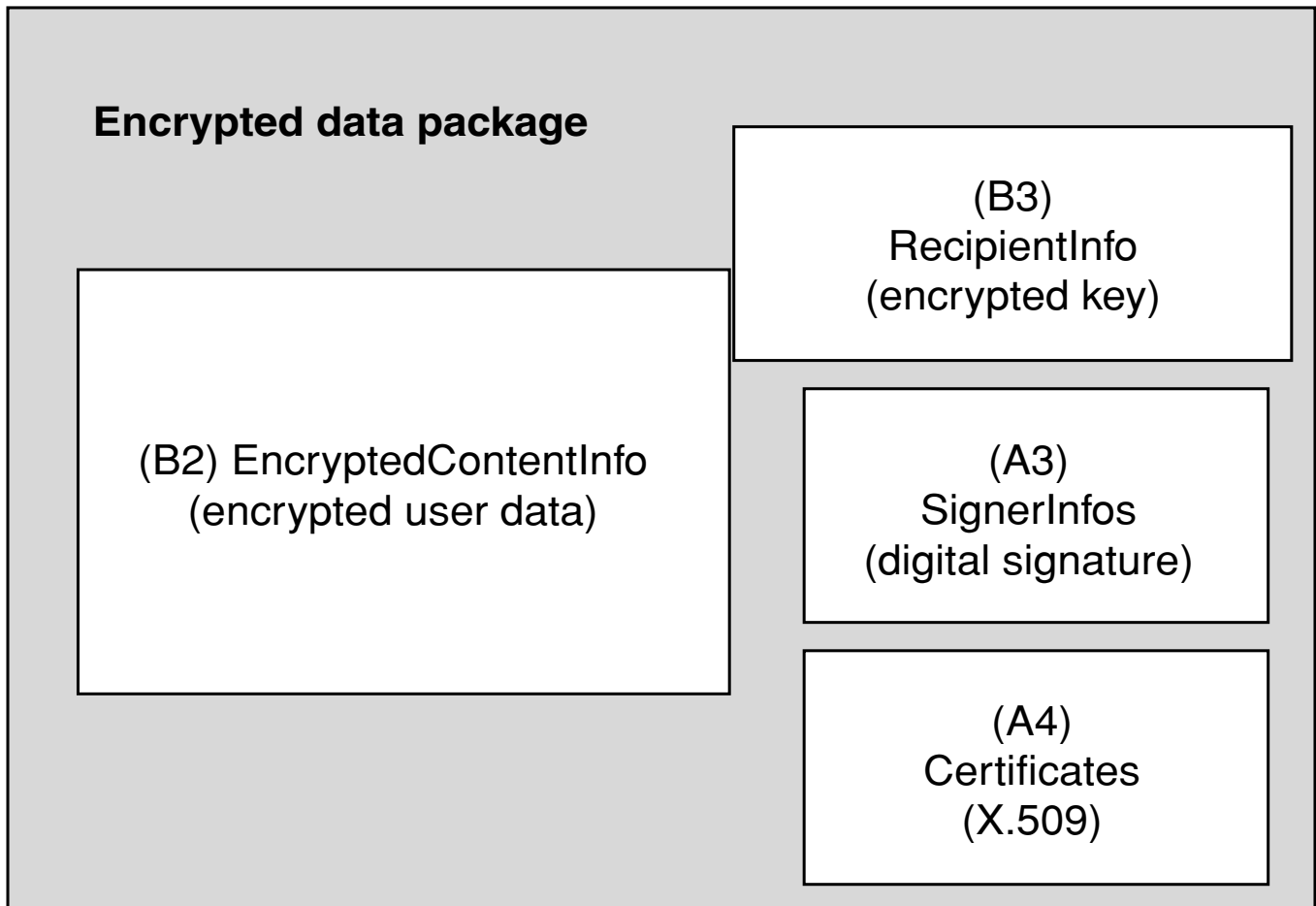
The purpose of the signature is to ensure the combination and contents of the individual data items, and to enable the recipient to detect any unauthorised alterations during transmission. Encryption is to prevent the unauthorised interception of data during communication.

The digital signature for the wrapped and signed package may be produced by either the applicant or his representative. The person that starts the transmission produces the digital signature for the final signed and encrypted package.





**(B1) SignedAndEnvelopedData <top level>**  
**(PKCS#7 digital envelope for transmission)**



Rules for producing the PKCS#7 digital envelope for transmission

**Table B1 SignedAndEnvelopedData – top level**

No.	Item name	PKCS#7 item	Content
1	Version	Version	Set integer value '1'
2	Recipient information	RecipientInfos	Set ONE <b>set</b> of RecipientInfo only (see table B3)
2	Set of algorithm identifiers	DigestAlgorithms	
2.1	Algorithm identifier	AlgorithmIdentifier	Set ONE <b>set</b> of algorithm identifiers {sha-1} only
3	Encrypted Content information	EncryptedContentInfo	Set one EncryptedContentInfo (see table B2)
4	Certificates	Certificates	Set one Certificates (see table A4)
5	Certificate revocation lists	Crls	Not used (set no data)
6	Signer information	SignerInfos	Set one SignerInfos (see table A3)

**Table B2 EncryptedContentInfo** – top level

No.	Item name	PKCS#7 item	Content
1	Content type	ContentType	Set object identifier {pkcs-7 1}
2	Content encryption algorithm	ContentEncryptionAlgorithm	Algorithm OBJECT identifier of content encryption (recommended algorithm: dES-EDE3-CBC)
3	Encrypted content	EncryptedContent	Encrypted user data

**Table B3 RecipientInfo** – top level

No.	Item name	PKCS#7 item	Content
1	Version	Version	Set integer value '1'
2	Issuer and serial number	IssuerAndSerialNumber	Issuer and serial number of certificate including public key for encrypting user data encryption key
3	Key encryption algorithm	KeyEncryptionAlgorithm	Algorithm OBJECT identifier for encrypting user data encryption key (recommended algorithm: rsaEncryption)
4	Encrypted key	EncryptedKey	Encrypted decryption key for user data

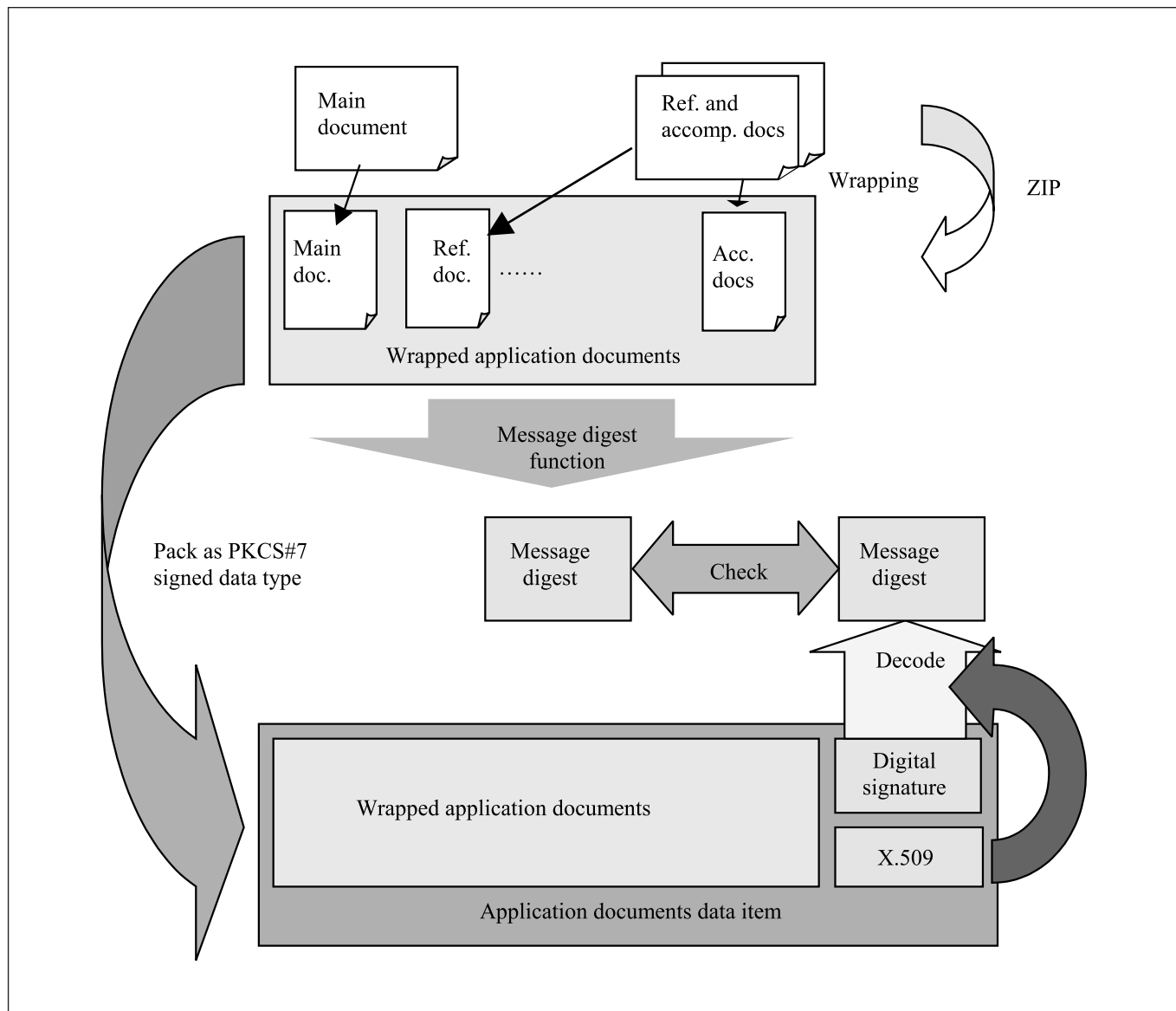
**6.2 Transmission mechanism**

The transmission mechanism operates as follows:

- An electronic session is established between the applicant and the EPO.
- The applicant transmits the signed and encrypted package.
- When the signed and encrypted package is received, its contents are checked for the presence of viruses and the

wrapped application documents object is processed to create its unique message digest.

- This digest is compared with the message digest included in the wrapped and signed package. If they match, an acknowledgement of receipt is sent to the applicant. If they do not, the applicant is informed accordingly. The session is then ended.



**6.2.1 Checking the message digest**

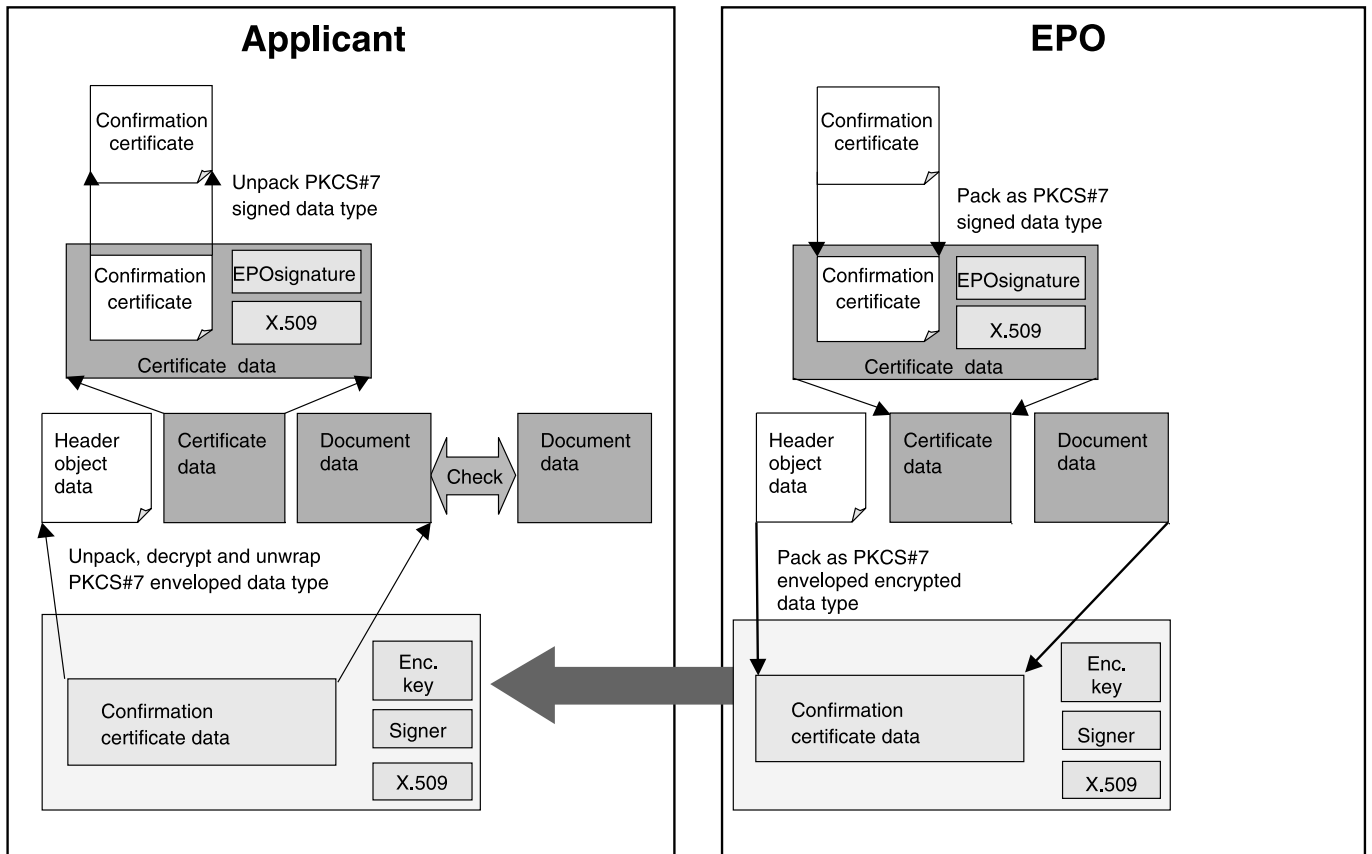
When the EPO receives the wrapped application documents, it opens the data items in them and ascertains the role of each one according to the information in the header object.

In the event of a communications or message digest comparison problem, the confirmation certificate contains information about the problem.

**6.2.2 Confirmation certificate**

The confirmation certificate data item includes a certificate data item, a header object data item indicating that the corresponding packet is a confirmation certificate, and, optionally, the application documents data item received with the new application.

The confirmation certificate is packaged as a signed and encrypted package, as described above.



The confirmation certificate is used to inform the applicant of the receipt of the application and must contain an XML version of this information. It may also contain a formatted version of the data in PDF. These files are combined in a single ZIP file and signed using the EPO's digital certificate.

**6.3 Transmission protocol**

The EPO uses a transfer protocol based on HTTP in conjunction with SSL.

**7 Physical media**

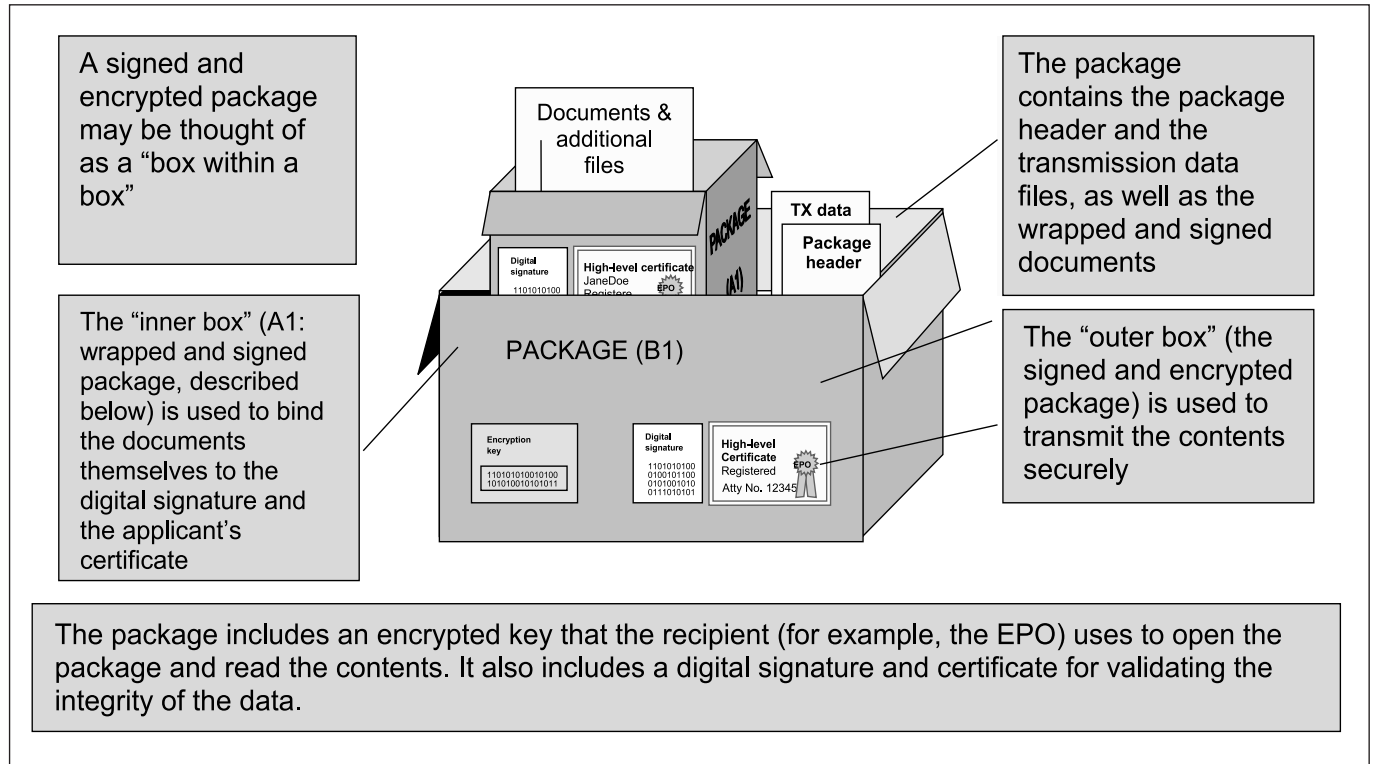
The EPO also accepts electronic filing on CD-R. Each CD-R should contain one application only, in the form of a signed WAD written into the root directory. The name of the signed WAD file should be "WAD.ZIP". The accompanying paper form should include details of the application or document and should refer to the "WAD.ZIP" file on the CD-R. The CD-R volume name should be based on the applicant's reference number.

**Annex – Diagrams illustrating the standard**

The following diagrams and text provide additional (simplified) information about the standard.

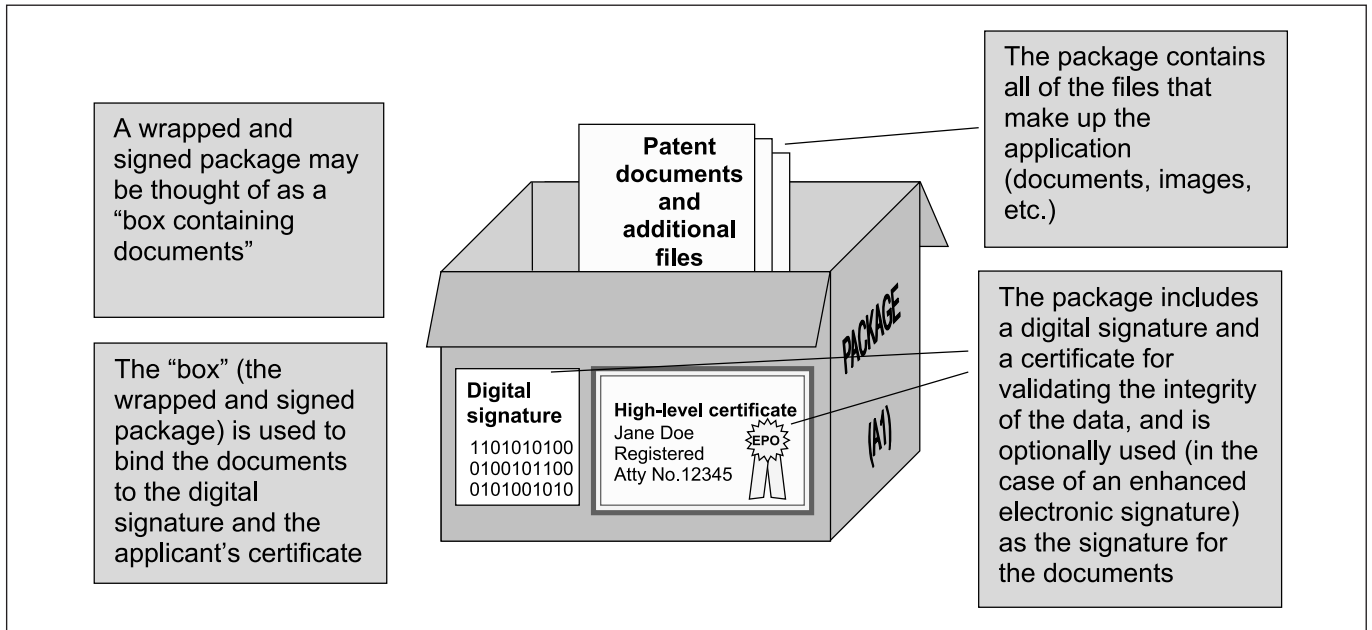
**Simplified anatomy of a signed and encrypted package**

Figure 1 illustrates, for non-technical readers, the components of the signed and encrypted package mechanism specified in this standard. The diagram has been intentionally simplified to obscure technical detail that may distract the reader from the key issues of the package design. For example, the ZIP wrapping has been left out, and encoding standards for objects are not addressed.



**Figure 1: Signed and encrypted package**

**Simplified anatomy of a wrapped and signed package**



**Figure 2: Wrapped and signed package**

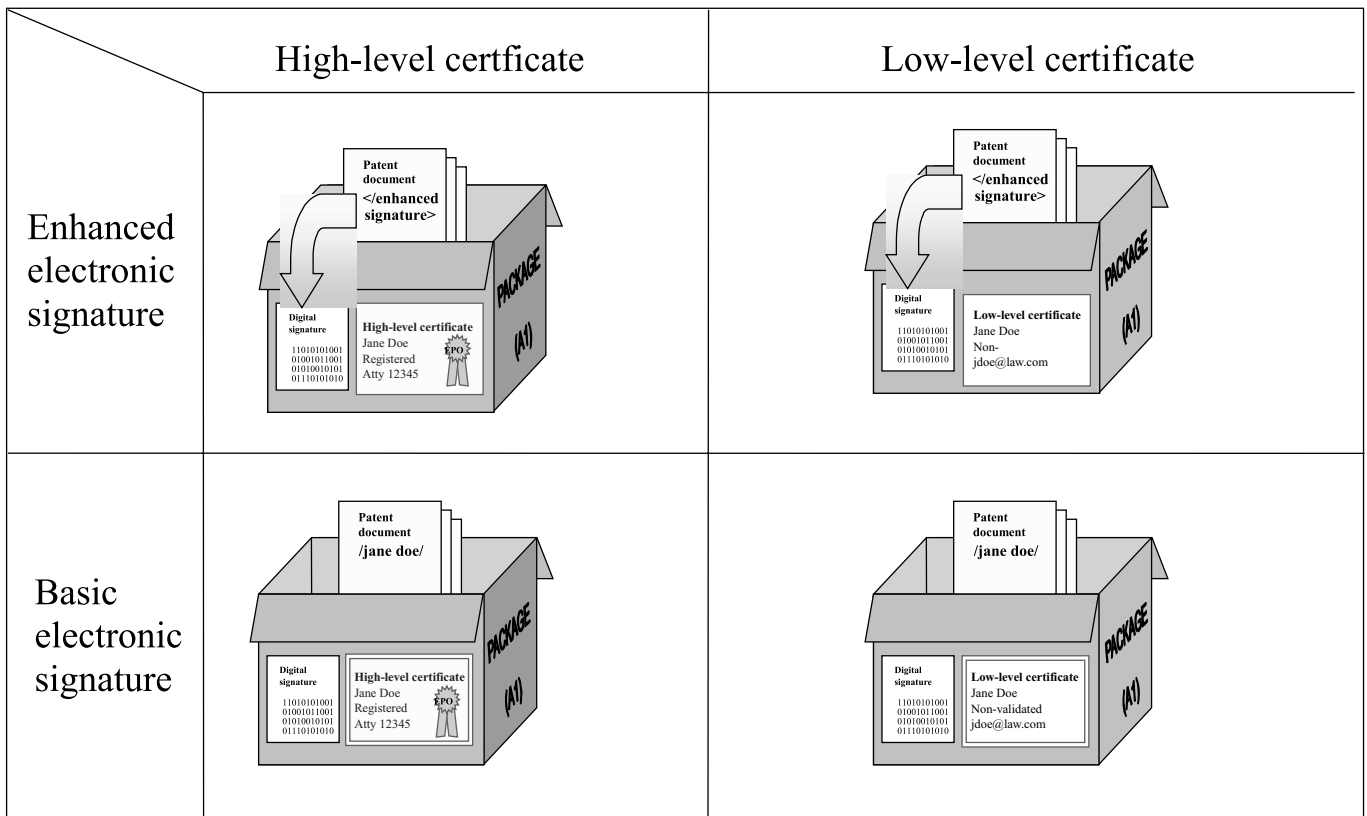
been 'zipped' together into a single file and placed in the root directory of the physical media.

**Anatomy of the wrapped application documents object**

The wrapped application documents object in section 5 defines how documents are "wrapped" together. In the case of offline submission on physical media, the further steps of creating the wrapped and signed package and the signed and encrypted package are optional. A wrapped application documents object consists of files that have

**Certificate/signature types**

The diagrams in Figures 3 to 7 illustrate the differences between the types of "digital certificate" and "electronic signature" options as specified in the standard. Each diagram shows a "box" representing the wrapped and signed package.



**Figure 3: Certificate/signature types**

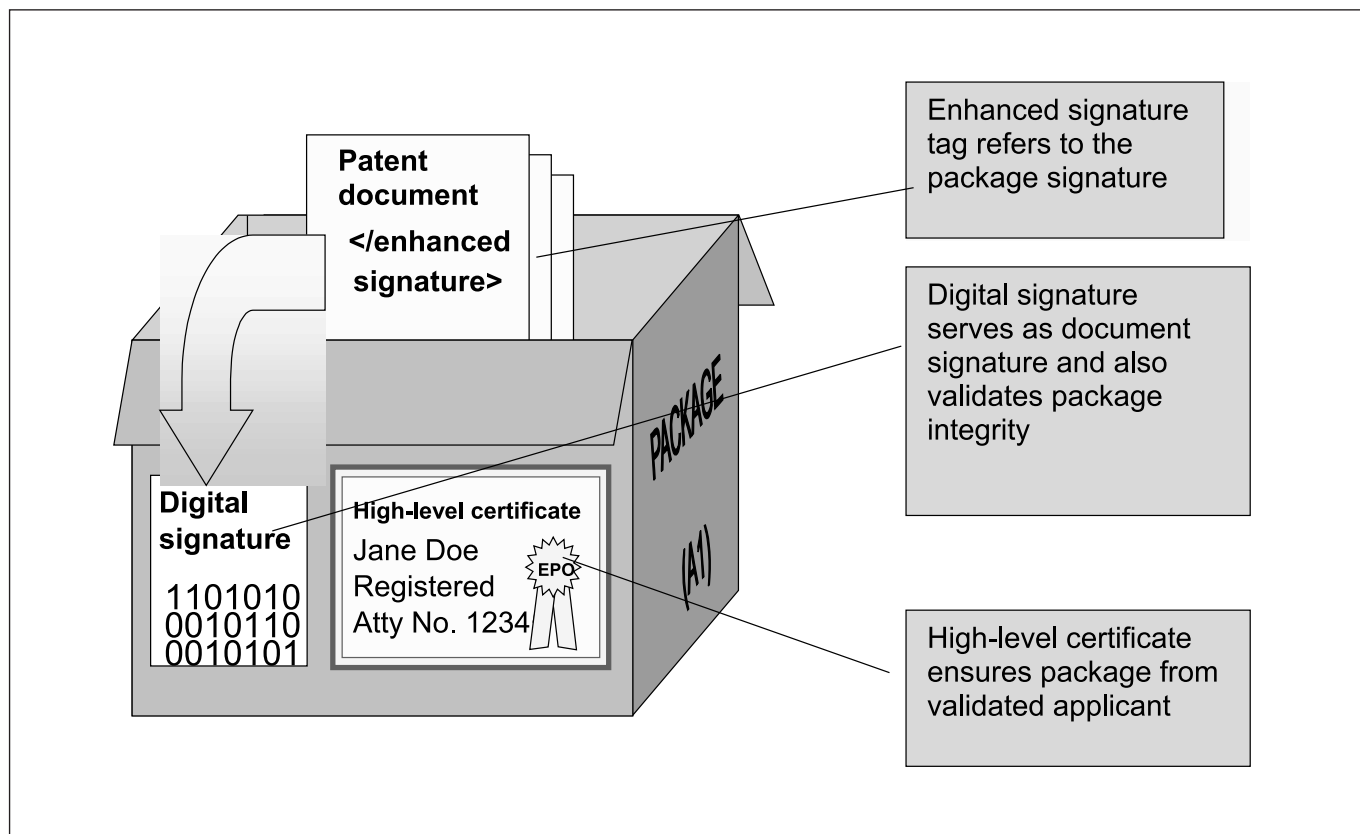


Figure 4: Enhanced electronic signature/high-level certificate

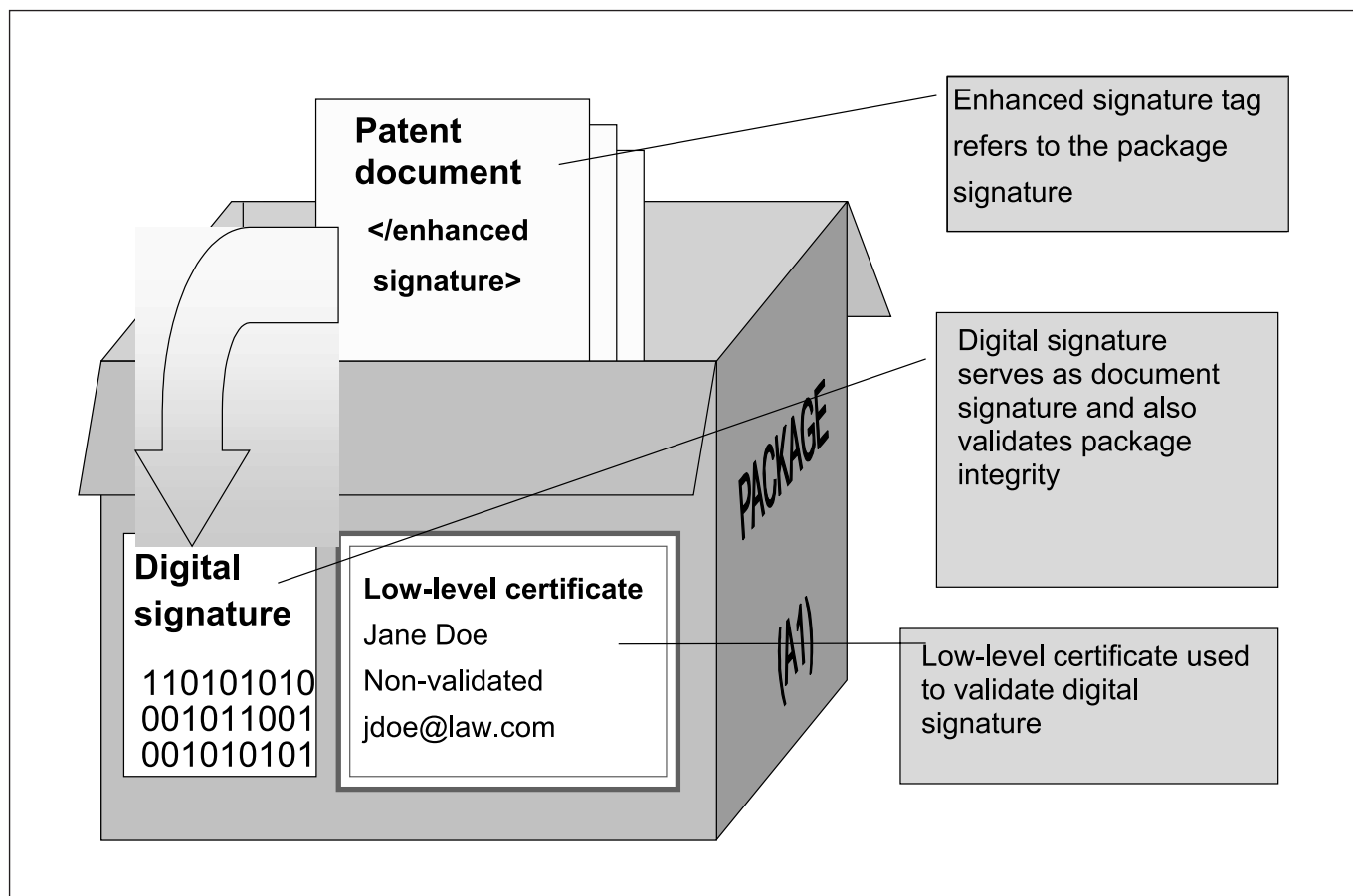


Figure 5: Enhanced electronic signature/low-level certificate

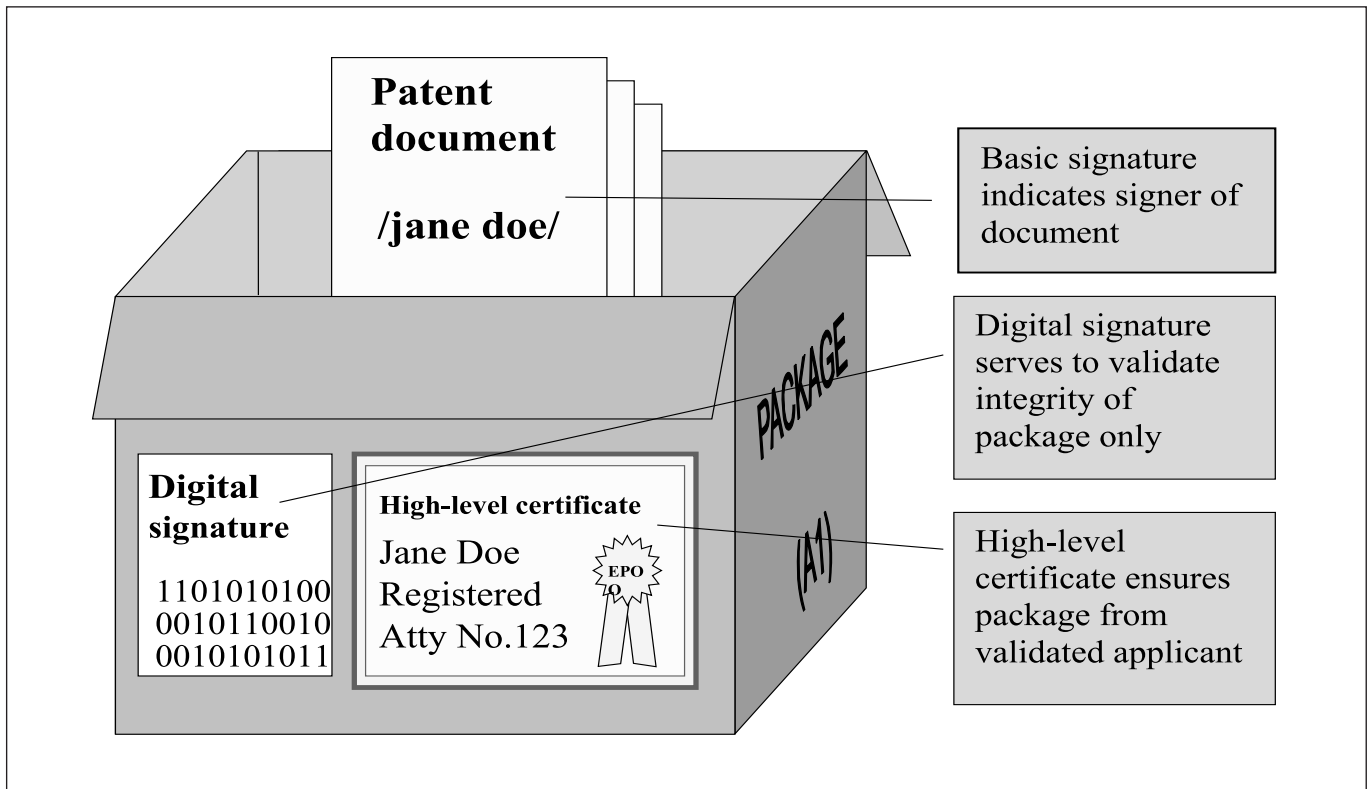


Figure 6: Basic electronic signature/high-level certificate

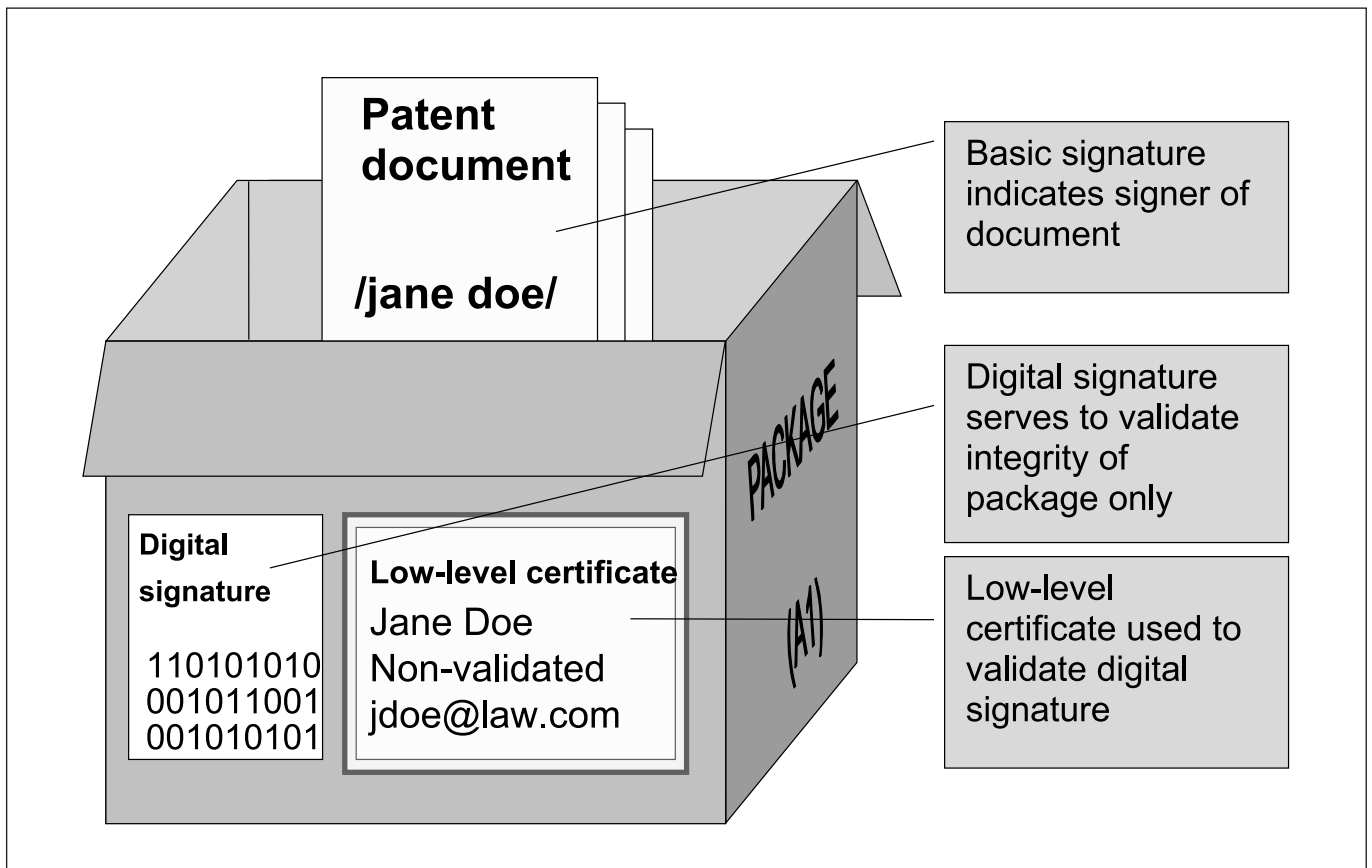


Figure 7: Basic electronic signature/low-level certificate



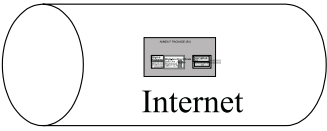








**Transmission mechanism/packaging combinations**

Figure 8 shows the various transmission mechanism/packaging combinations that are permissible. The following applies to each transmission mechanism:

(a) Online/internet: a signed and encrypted package must be used.

(b) Online/secure (channel encryption such as a private network): a signed and encrypted package or wrapped and signed package must be used.

(c) Offline/physical media: either a signed and encrypted package, a wrapped and signed package, or a wrapped application documents package may be used.

	Signed and encrypted package	Wrapped and signed package	Wrapped application documents
Online/ Internet	 Internet	 Not permitted	 Not permitted
Online/ Secure	 Secure	 Secure	 Not permitted
Offline media			

**Figure 8: Transmission protocols and packages permitted**