INTERNATIONAL TELECOMMUNICATION UNION

CCITT T.410 Series

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

TERMINAL EQUIPMENT AND PROTOCOLS FOR TELEMATIC SERVICES

FIRST EXTENSION (JANUARY 1991) TO THE T.410 SERIES (1988) OF RECOMMENDATIONS CONTAINED IN THE CCITT BLUE BOOK, FASCICLE VII.6:

I – Tiled raster graphics
II – Annex E to T.411
III – Alternative representation
IV – Styles extension
V – Security

Recommendations: T.410 Series

Geneva, 1991

Printed in Switzerland

FOREWORD

The CCITT (the International Telegraph and Telephone Consultative Committee) is a permanent organ of the International Telecommunication Union (ITU). CCITT is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

The first Extension (January 1991) to the T.410 Series (1988) was prepared by Study Group VIII and was approved under the Resolution No. 2 procedure on the 18 of January 1991.

CCITT NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication Administration and a recognized private operating agency.

© ITU 1991

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

PAGE BLANCHE

I Addendum to the T.410-Series of Recommendations (1988) on the subject of tiled raster graphics

Add the following definition after § 3.167 of Recommendation T.411.

3.168 *Tile*

One element of a two dimensional array of non-overlapping rectangular regions of a pel array.

Add the following paragraphs after § 5.3.2 (Discarded pels) of Recommendation T.417.

5.4 *Tiling*

The pel array may be segmented into a two dimensional array of non-overlapping rectangular regions called tiles. The content information of each tile is coded independently of the content information of the other tiles of the same pel array.

Tiling facilitates convenient access to, and/or processing of portions of the pel array independent of access to, and/or processing of other portions. The capability to encode each tile separately as bitmap, compressed or null maximizes the possible compression of the tiled pel array.

Note – Tiling provides an alternative method for coding of raster graphics content and therefore does not affect the positioning of the clipped pel array.

The basic concepts, pel image model and positioning of pels in a basic layout object continue to apply. Furthermore the attributes for pel array clipping continue to apply to the pel array.

The location of pel array content relative to the tile content is specified by the “tiling offset” attribute. Figure 3 illustrates the location of the pel array in the set of tiles.

Tiled raster graphics content information consists of a sequence of tiles ordered in the direction of the pel path and line progression as illustrated in Figure 4.

The content of each tile may be T.4, T.6 or bitmap encoded as specified by the coding attributes. Alternatively, it may be omitted if the pels within the tile are either all foreground or all background.

FIGURE 3 = 11,5 cm

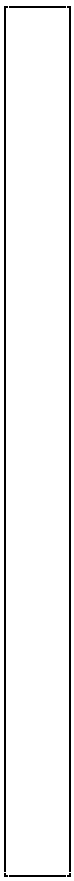
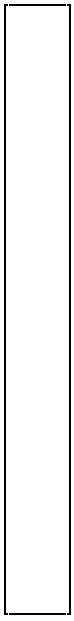


FIGURE 4 = 8 cm



Renumber subsequent paragraphs to take account of the insertion of the new § 5.4.

In the second paragraph of § 5 (Principles of positioning pels) of Recommendation T.417, *replace*

. . . in § 5.4.1. Paragraphs 5.4.2 and 5.4.3 then . . .

by

. . . in § 5.5.1. Paragraphs 5.5.2 and 5.5.3 then . . .

In the third paragraph of what was § 5.4.1 (Positioning parameters) of Recommendation T.417, *replace*

. . . in §§ 5.4.2 and 5.4.3 respectively

by

. . . in §§ 5.5.2 and 5.5.3 respectively.

Add the following to the end of the permissible value list in § 7.1.1 (Type of coding) of Recommendation T.417.

{2 8 3 7 5} for "tiled encoding".

Add the following to the end of the list of dashed items in the definition area of clause 7.1.1 (Type of coding) of Recommendation T.417.

– "tiled encoding" according to the tiling scheme defined herein, the bitmap encoding scheme, the two dimensional encoding scheme defined in Recommendation T.6, or the one or two dimensional encoding scheme defined in Recommendation T.4.

Add the following paragraph after the line beginning "An explanation of these coding schemes . . ." in § 7.1.1 (Type of coding) of Recommendation T.417.

The value "tiled encoding" indicates that the tiles in the content portion are each encoded per the value of the associated "tile types" attribute as defined in § 7.2.8.

Add the following paragraph to the definition of § 7.1.1 (Type of coding) of Recommendation T.417.

The relationship between the order of pels, the order of encoded bits and the order of encoded octets is the same for tiled, as for untiled bitmap, T.4 and T.6 encoding.

Add the following text at the end of the last sentence of § 7.2.1 of Recommendation T.417.

. . . or "tiled encoding".

Add the following four paragraphs after § 7.2.4 (Number of discarded pels) of Recommendation T.417.

7.2.5 *Number of lines per tile*

Classification: Defaultable;

Applicability: Formatted processable content architecture class;

Permissible value: Positive integer;

Default value: 512

Definition:

This attribute specifies the tile dimension in units of "line spaces" in the direction of line progression.

This attribute is only applicable if the value of the attribute "type of coding" is "tiled encoding".

7.2.6 *Number of pels per tile line*

Classification: Defaultable;

Applicability: Formatted processable content architecture class;

Permissible value: Positive integer;

Default value: 512

Definition:

This attribute specifies the tile dimension in units of “pel spaces” in the pel path direction.

This attribute is only applicable if the value of the attribute “type of coding” is “tiled encoding”.

7.2.7 *Tiling offset*

Classification: Defaultable;

Applicability: Formatted processable content architecture class;

Structure: Coordinate pair: X coordinate; Y coordinate.

Permissible values: Coordinate pair:

non-negative integer less than “number of pels per tile line” non-negative integer less than“number of lines per tile”;

Default value: (0,0);

Definition:

This attribute specifies the location of the pel array within the tile space by defining the offset of the first pel of the pel array from the first pel position of the first tile. The offset is specified in pel spaces in the direction of the pel path and line spaces in the direction of the line progression.

All tiles cover a portion of the pel array. Portions of the tile space outside the pel array are artifacts of tiling and contain no information.

This attribute is only applicable if the value of the attribute “type of coding” is “tiled encoding”.

7.2.8 *Tile types*

Classification: Defaultable;

Applicability: Formatted processable content architecture class;

Permissible values: A sequence of one or more data elements with one of the following values:

“null background”, “null foreground”, “bitmap encoded” “T.6 encoded” “T.4 one dimensional encoded” “T.4 two dimensional encoded”;

Default value: All tiles are T.6 encoded.

Definition:

This attribute indicates the types of coding of tiles in the content portion as a sequence of values. Each value specifies the type of coding of the corresponding tile (see Figure 4) in the content portion as follows:

– “null background”, indicating that all pels in the tile are known to be background and the tile has no encoded content;

– “null foreground”, indicating that all pels in the tile are known to be foreground and the tile has no encoded content;

– “T.6 encoded”, indicating that the pels in the tile are encoded as a T.6 octet string;

– “T.4 one dimensional encoded”, indicating that the pels in the tile are encoded as a T.4 one dimensional octet string;

– “T.4 two dimensional encoded”, indicating that the pels in the tile are encoded as a T.4 two dimensional octet string;

– “bitmap encoded”, indicating that the pels in the tile are encoded as a bitmap octet string.

The number of values is equal to the number of tiles.

This attribute is only applicable if the value of the attribute “type of coding” is “tiled encoding”.

In § 8.2 of Recommendation T.417, *replace*

```
EXPORTS Raster-Graphics-Attributes,
        One-Of-Four-Angles,
        One-Of-Two-Angles,
        Pel-Transmission-Density,
        Measure-Pair,
        Clipping,
        Pel-Spacing,
        Spacing-Ratio,
        Image-Dimensions;
```

by

```
EXPORTS Raster-Graphics-Attributes,
        One-Of-Four-Angles,
        One-Of-Two-Angles,
        Pel-Transmission-Density,
        Pel-Spacing,
        Spacing-Ratio,
        Coordinate-Pair;
```

In § 8.3 of Recommendation T.417, *replace*

```
EXPORTS Raster-Gr-Coding-Attributes,
        Compression;
```

by

```
EXPORTS Raster-Gr-Coding-Attributes,
        Compression, Tile-Type;
```

Add the following below the list of EXPORTS in § 8.3 (Representation of coding attributes) of Recommendation T.417.

```
IMPORTS Coordinate-Pair
        FROM Raster-Gr-Presentation-Attributes;
```

Add the following to “Raster-Gr-Coding-Attribute” in § 8.3 (Representation of coding attributes) of Recommendation T.417.

```
number-of-pels-per-tile-line    [6] IMPLICIT INTEGER OPTIONAL,
number-of-lines-per-tile        [7] IMPLICIT INTEGER OPTIONAL,
tiling-offset                   [8] IMPLICIT Coordinate-Pair
                                            OPTIONAL,
tile-types                      [9] IMPLICIT SEQUENCE OF Tile-Type
                                            OPTIONAL
```

Add the following after the "Compression" definition in § 8.3 (Representation of coding attributes) of Recommendation T.417.

```
Tile-Type                          ::= INTEGER {
                                       null-background                (0),
                                       null-foreground                (1),
                                       T.6-encoded                    (2),
                                       T.4-one-dimensional-encoded    (3),
                                       T.4-two-dimensional-encoded    (4),
                                       bitmap-encoded                 (5)
    }
```

In § 8.4 of Recommendation T.417 *replace*

```
IMPORTS
        One-Of-Four-Angles,
        One-Of-Two-Angles,
        Pel-Transmission-Density,
        Measure-Pair,
        Clipping,
        Pel-Spacing,
        Spacing-Ratio,
        Image-Dimensions,
        FROM Raster-Gr-Presentation-Attributes,
        Compression,
        FROM Raster-Gr-Coding-Attributes;
```

by

```
IMPORTS
        One-Of-Four-Angles,
        One-Of-Two-Angles,
        Pel-Transmission-Density,
        Pel-Spacing,
        Spacing-Ratio,
        Coordinate-Pair,
        FROM Raster-Gr-Presentation-Attributes,
        Compression,
        Tile-Type,
        FROM Raster-Gr-Coding-Attributes;
```

In § 8.4 (Representation of non-basic features and non-standard defaults) of Recommendation T.417, *replace*

```
        compression                [0] IMPLICIT Compression }
```

by

```
        compression                [0] IMPLICIT Compression,
        number-of-pels-per-tile-line   [6] IMPLICIT INTEGER,
        number-of-lines-per-tile   [7] IMPLICIT INTEGER,
        tiling-offset              [8] IMPLICIT Coordinate-Pair,
        tiling-types               [9] IMPLICIT Tile-Type }
```

In § 8.4 (Representation of non-basic features and non-standard defaults) of Recommendation T.417, *replace*

```
        compression                [8] IMPLICIT Compression
                                       OPTIONAL }
```

by

compression **[8] IMPLICIT Compression OPTIONAL,**
number-of-pels-per-tile-line **[11] IMPLICIT INTEGER OPTIONAL,**
number-of-lines-per-tile **[12] IMPLICIT INTEGER OPTIONAL,**
tiling-offset **[13] IMPLICIT Coordinate-Pair OPTIONAL,**
tiling-types **[14] IMPLICIT Tile-Type OPTIONAL }**

In § 9 (Coding schemes) of Recommendation T.417, *replace*

– bitmap encoding scheme.

by

– bitmap encoding scheme;

– tiled encoding scheme.

Add the following paragraph after § 9.3 (Bitmap encoding scheme) of Recommendation T.417.

9.4 *Tiled encoding scheme*

Tiled content information is coded as a structured octet string composed of a sequence of independently coded tiles. Each tile is encoded as one octet string which may be structured or unstructured.

All the pels of a tile may be coded according to one of the following coding schemes:

– Group 4 facsimile encoding scheme;

– Group 3 facsimile encoding schemes;

– bitmap encoding scheme.

Alternatively the pels of a tile may be all background or all foreground, and not be coded.

Add the following to Table 6 in § 12 (Definition of raster graphics content architecture classes) of Recommendation T.417.

Number of pels per tile line	–	D 3)
Number of lines per tile	–	D 3)
Tiling offset	–	D 3)
Tile types	–	D 3)

Note 3 – This attribute is only applicable if the value of the attribute "type of coding" is "tiled encoding".

Add the following to § A.2.2 of the annexes of Recommendation T.417.

[Type of coding]	Tiled encoding	
Number of pels per tile line	Any positive integer	512
Number of lines per tile	Any positive integer	512
Tiling offset Coordinate Pair	(Any non-negative integer, less than "number of pels per tile line", Any non-negative integer less than "number of lines per tile")	(0,0)
Tile types	Sequence of positive integers	compressed as in T.6

In § A.2.2 of the annexes of Recommendation T.417, *replace*

Note – The attribute "compression" is only applicable if the value of the attribute "type of coding" is "T.6 encoding" or "T.4 two dimensional encoding".

by

Note – The attribute "compression" is only applicable if the value of the attribute "type of coding" is "T.6 encoding", "T.4 two dimensional encoding" or "tiled encoding".

II **Revision of Recommendation T.411** (Reference List and Annex E)

II.1 *Reference List:*

- Recommendation T.414 (1988): Open document architecture (ODA) and interchange format – Document profile;
- Recommendation T.415 (1988): Open document architecture (ODA) and interchange format – Open document interchange format (ODIF);
- Recommendation T.416 (1988): Open document architecture (ODA) and interchange format – Character content architectures;
- Recommendation T.417 (1988): Open document architecture (ODA) and interchange format – Raster graphics content architectures;
- Recommendation T.418 (1988): Open document architecture (ODA) and interchange format – Geometric graphics content architecture;

- Recommendation X.411 (1988): Message handling systems: Message transfer system: Abstract service definition and procedures;
- Recommendation X.420 (1988): Message handling systems: Interpersonal messaging system;
- ISO 2022 (1986): Information processing – ISO 7-bit and 8-bit coded character sets – Code extension techniques;
- ISO 6429 (1988): Information processing Control functions for 7-bit and 8-bit coded character sets;
- ISO 6937-1 (1983): Information processing – Coded character sets for text communication – Part 1: General introduction;
- ISO 6937-2 (1983): Information processing – Coded character sets for text communication – Part 2: Latin alphabetic and non-alphabetic graphic characters;
- ISO 6937-3 (1983): Information processing – Coded character sets for text communication – Part 3: Control functions for page-image format;[1)]
- ISO 8601 (1988): Data elements and interchange formats – Information interchange – Representation of dates and times;[2)]
- ISO 8632-1 (1987): Information processing systems – Computer graphics – Metafile for the storage and transfer of picture description information – Part 1: Functional specification;
- ISO 8632-3 (1987): Information processing systems – Computer graphics – Metafile for the storage and transfer of picture description information – Part 3: Binary encoding;
- ISO 9541-1: Information processing – Font and character information interchange – Part 1: Introduction[2)];
- ISO 9541-2: Information processing – Font and character information interchange – Part 2: Registration and naming procedures[1)];
- ISO 9541-5: Information processing – Font and character information interchange – Part 5: Font attributes and character model[1)];
- ISO 9541-6: Information processing – Font and character information interchange – Part 6: Font and character attribute subsets and applications[1)].

II.2

ANNEX E

(to Recommendation T.411)
(Normative)

Use of MHS to interchange documents conforming to the T.410-Series of Recommendations

E.1 *ODA identification in the P.1 Protocol of MHS*

Documents shall be identified by a set of ASN.1 object identifiers as externally-defined encoded-information types. One member shall always be the ASN.1 object identifier for ODA, the other members shall be one or more ASN.1 object identifiers for the document application profiles to which the message body parts conform.

ODA document	{2800 }
Document application profile	{see Note 2}
. . . .	{ . . . . }
. . . .	{ . . . . }
. . . .	{ . . . . }

1) Under revision.

2) To be published.

Note 1 – When using [MHS/MOTIS] to transfer documents conforming to ODA, the MTS may perform format conversion. Format conversion of ODA documents may result in loss of information. If format conversion is not appropriate, this should be indicated by the sender when submitting a message with ODA body parts to [MHS/MOTIS].

Note 2 – These document application profiles ASN.1 object identifiers are those defined for CCITT. Other organizations shall use object identifiers as appropriate.

E.2 *ODA identification in the P.2 protocol of MHS*

Documents conforming to ODA shall be identified as ODA extended body parts. Each extended body part shall contain parameter information about the applicable document application profile and the document architecture class.

Note – ODA body parts can be mixed with non-ODA body parts in a P.2 body.

The module for specifying the ODA body parts is described below:

```
IPMSExtendedBodyPartTypeOda {joint-iso-ccitt(2) oda(8) modules(1)
    part(0) extended-body-part-type-oda(0) }

Definitions implicit tags ::=

Begin
-- prologue

EXPORTS
    oda-body-part,
    OdaBodyPartParameters,
    OdaData;

IMPORTS
    Interchange-Data-Element,
    FROM Interchange-Data-Elements {2 8 1 5 5},
    EXTENDED-BODY-PART-TYPE,
    FROM IPMSInformationObjects {joint-iso-ccitt(2)
        mhs-motis(6) ipms(1) modules(0)
        information-objects(2) };

oda-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS    OdaBodyPartParameters  IDENTIFIED BY id-et-oda-param
    DATA          OdaData
    ::= id-et-oda-data

-- Abstract syntax for ODA bodypart parameters shall appear in the parameter elements of an IPM
ExternallyDefinedBodyPart --

OdaBodyPartParameters                ::= SET {
    document-application-profile     [0] OBJECT IDENTIFIER

-- This object identifier value shall also be used in the MTS ExternalEncodedInformationType in addition to
the id-et-oda-data object identifier --,

    document-architecture-class      [1] INTEGER {
                                         formatted (0),
                                         processable (1),
                                         formatted processable (2) }}
```

-- Abstract syntax for ODA data shall appear in the data element of an IPM ExternallyDefinedBodyPart --

OdaData ::= SEQUENCE OF
Interchange-Data-Element

id-et-oda-param OBJECT IDENTIFIER ::= {2 8 1 1 2}

-- Identifies the Abstract Syntax for ODA bodypart parameters using the ASN.1 basic encoding rules --,

id-et-oda-data OBJECT IDENTIFIER ::= {2 8 1 1 1}

-- Identifies the Abstract Syntax for ODA data using the ASN.1 basic encoding rules --

END -- of IPMSExtendedBodyPartTypeOda --

III Addendum to the T.410-Series of Recommendations (1988) on the subject of alternative representation

[*Recommendation T.411/Part 1*]

Add to the definitions the clause of [Recommendation T.411/Part 1]:

- *alternative description:* A description that represents a basic object that is intended to be used by the recipient in lieu of the primary description of that basic object when the primary description cannot be processed;
- *alternative subtree:* An alternative basic object description in conjunction with its associated content portion description, if any;
- *primary description:* A description that represents a basic object that best represents the intent of the originator;
- *primary subtree:* The basic object description in conjunction with its associated content portion descriptions, with the potential to be replaced by an alternative subtree.

Add to the definition of description:

Note – A basic object may have several descriptions when alternative descriptions are used.

[*Recommendation T.412/Part 2*]

Add to the end of § 2.3.3:

An object description may also be referred to as a *primary object description,* in particular when it is required to distinguish between object descriptions and *alternative descriptions* [see § 2.3.5a)].

Add after § 2.3.5 (Styles):

2.3.5a) *Alternative descriptions*

In addition to its primary description, a basic logical object or a basic layout *object* may be represented by one or more *alternative descriptions*. An alternative description is intended by the originator to be used by the recipient of a document in lieu of the primary description when the recipient is not capable of processing the primary description. In the case that these are multiple alternative descriptions, a preference order is defined between these alternative descriptions. An alternative description for a basic object may specify the same or a different set of attributes as those specified by the primary description.

Alternative descriptions provide fallback mechanisms in this International Standard. Possible uses include: fallback content architecture classes (e.g., providing a raster image as a fallback for geometric graphics), compatibility with several versions of ODA, compatibility with several document application profiles, and fallback for non-basic values within a document application profile.

Using an alternative description for an object implies using a different set of associated content portion descriptions, if any. An alternative description for an object in conjunction with any associated content portion description is collectively called an *alternative subtree*. The basic object description in conjunction with its content portion descriptions which could be replaced by an alternative subtree is called the *primary subtree*.

A fallback for a content portion description can be supplied by providing an alternative description for a basic object with which the content portion is associated, and associating the fallback content portion description with this alternative basic object description.

Alternative subtrees must satisfy the constraint that substituting the alternative subtree for its primary subtree must result in a valid document for the purposes of performing a layout process and/or an imaging process.

Add at the end of § 2.5.1 (Editing process):

2.5.1.8 *Alternative descriptions*

Alternative descriptions usually play no direct role in the editing process but may be derived automatically by the originating system from the primary descriptions. It is then the responsibility of the originating system to keep the alternative descriptions consistent with the primary description. Alternative descriptions may also have to be manually derived from the primary description (for example, descriptive text indicating the contents of the primary description); in this case it is the responsibility of the originator to maintain consistency during the editing process.

At the end of § 2.5.2 (Layout process):

2.5.2.9 *Alternative descriptions*

Alternative descriptions do not influence the reference layout process. If a system is to lay out a document that contains primary descriptions it is not capable of processing, it may substitute alternative descriptions for those primary descriptions prior to the layout process.

A system that provides a layout process where both primary and alternative descriptions influence the layout process, is outside the scope of this International Standard.

Add at the end of § 2.5.3 (Imaging process):

2.5.3.6 *Alternative descriptions*

Alternative descriptions do not influence the reference imaging process. If a system is to image a document that contains primary descriptions it is not capable of processing, it may substitute alternative descriptions for these primary descriptions prior to the imaging process.

§ 5.3.3.2 *Subordinates*

Add at the end of the first paragraph of definition of Subordinates:

In the case of subordinate basic objects this attribute identifies their primary descriptions.

Add a new clause 5.3.3x (Alternative):

5.3.3x *Alternative*

Constituent: Basic object descriptions

Classification: Non-mandatory

Permissible values: The identifier of an alternative description of a basic object

Definition:

For a primary description of an object this attribute refers to the first alternative description, in the order of preference. For an alternative description this attribute refers to the next alternative description, in the order of preference.

Note – Thus, the specification of this attribute establishes a chain of alternative descriptions to a primary description of a basic object in the order of decreasing preference.

Add a new § 5.3.3y (Primary):

5.3.3y *Primary*

Constituents: Basic object descriptions. Applicable only to alternative descriptions

Classification: Mandatory for alternative descriptions

Permissible values: The identifier of a basic object

Definition:

This attribute refers from an alternative description to its primary description.

Add to the end of the first paragraph in § 6.1:

If necessary, the logical structure is derived from the set of descriptions, including alternative descriptions present in the document, by a process called initialization [see § 6.1a)].

Add a new § 6.1a):

6.1a *Initialization*

Before the layout process commences, the logical structure of the document is derived conceptually from the set of primary and alternative descriptions in the document by the following initialization of the layout process.

First, a logical structure is created from the primary descriptions in the document, i.e., the root logical object description and all those descriptions which are referred to by the attribute "subordinates" of composite object descriptions. If any of the primary descriptions cannot be decoded by the recipient, the initialization can substitute one or more primary descriptions by one or more alternative descriptions.

If the resulting logical structure cannot be processed by the recipient (e.g., edited) or may result in layout that cannot be processed by the recipient, the initialization can substitute one or more primary or alternative descriptions by one or more alternative descriptions.

These substitutions take account of the preference order for substitution specified by the attribute "alternative".

Note – For a more implementation-oriented description of the initialization see Annex G (this information is informative rather than normative).

Add to the end of the first paragraph in § 7.1:

If necessary, the layout structure is derived from the set of descriptions including alternative descriptions present in the document by a process called initialization; this process is performed analogously to the initialization of the layout process [see § 6.1a)].

[*Recommendation T.414/Part 4*]

Add to the document profile:

5.3.6a *Alternative feature sets*

This attribute lists combinations of identified features, so that any one combination is sufficient to process a particular selection of primary descriptions and alternative descriptions in the document.

This attribute consists of a set of sets of ASN.1 object identifiers. Each set lists a set of object identifiers for features such as content architecture classes that is sufficient to process a particular set of alternatives in the document.

Various parts of this International Standard define ASN.1 object identifiers for features. In particular, content architectures specify an ASN.1 object identifier for each architecture class.

Note 1 – In the T.410-Series of Recommendations no other features are defined.

Note 2 – No provision is made for alternative sets of non-basic values.

[*Recommendation T.415/Part 5*]

Add at the end of § 5.2:

For basic objects for which alternative descriptions have been specified there is one descriptor representing the primary description and one descriptor for each alternative description. In the data stream, the descriptors for alternative descriptions of basic object descriptions follow immediately after the descriptors for their primary description, in the order of decreasing preference. The text units representing the content portions associated to alternative subtrees follow immediately after the text units representing the content portions associated to the primary subtree, in the order of decreasing preference.

Add in § 5.6 in the definition of "Document-Characteristics" after the lines "ODA-Version":

Alternative-feature-sets **[11] IMPLICIT SET OF**
SET OF OBJECT IDENTIFIER OPTIONAL

In §§ 5.8 and 5.9 in the definition of "Layout-Object-Descriptor-Body" and "Logical-Object-Descriptor-Body":

change the terminating brace into a comma, and *append*

primary **[26] IMPLICIT Object-or-class-identifier OPTIONAL,**
alternative **[27] IMPLICIT Object-or-class-identifier OPTIONAL}**

[*Recommendation T.412/Part 2*]

Add a new informative Annex G:

ANNEX G

(Informative)

Overview of alternative description, technical and implementation aspects

G.1 *Substituting basic objects*

The basic mechanism employed by alternative descriptions is the substitution of entire basic objects in an ODA document based on the presence or absence of capabilities of either the layout or the imaging process.

G.2 *Independence of substitutions*

All these substitutions are made independently of each other, i.e., the fact that a particular subtree is used instead of a different subtree would have no relation to the fact that at a different place in the document another subtree is used instead of a different subtree applicable at that point.

G.3 *Selection of alternatives*

The decision to use a primary subtree or an alternative subtree has been called *selection*.

Selection can happen at two conceptual places:

1) in the initialization phase of the layout process; and

2) in the initialization phase of the imaging process.

In both cases the actual implementation is likely to perform the selection during the respective process, but from a conceptual view it is preferable to think about selection taking place before the process is commenced. This allows use of the semantics of the ODA layout and imaging processes without any change.

G.4 *Substitution in the initialization process*

Once an alternate substitution has been made this substitution proceeds as follows: The primary description is ignored in the layout and imaging processes. All content portion descriptions associated with the primary description are ignored in the layout and imaging processes. An alternative description which refers to the primary description is processed instead. Its object identifier is changed to that of the primary description. In the content portion descriptions the matching change of identifiers is performed.

Such substitutions can be performed repeatedly. The alternative descriptions for each primary description are considered in the orders of preference specified by the values of the attribute “alternative”. If no logical structure can be created that can be processed by the recipient, the initialization process fails.

Implementations that perform the initialization process directly from an ODIF stream need not use the attribute “alternative”.

G.5 *Syntactical selection of alternatives*

Sometimes fallbacks may be required because a recipient system cannot even decode a constituent, e.g., because a new format for this constituent or even a new type of constituent is used. This means that only providing pointers to the alternative descriptions in the primary descriptions would have been contrary to the purpose of providing alternative descriptions, since the reason that the alternative description is needed may be that the primary cannot be syntactically understood.

An association of an alternative to a primary thus was made by identifying the alternative to be a substitute for the primary. This also means that a system trying to read a document cannot immediately give up upon not being able to decode a description but needs to continue reading for possible alternatives. To simplify this, [Recommendation T.415/Part 5] specifies that alternative descriptions immediately follow the primary description in

the interchange data stream. Because of this constraint on the sequential order of alternative descriptions in the interchange format it is not necessary to make use of the attribute “alternate” for finding alternative descriptions. It is also possible that a description that could not be parsed and for which no alternative is present is a class or style description used by an object for which an alternative is provided that uses a different style or class; in that case not being able to parse the style or class is not an error condition (this is in accordance with general robustness principles).

The resulting decoding strategy for ODIF documents could be called “read until you understand”: if a descriptor cannot be decoded the recipient should continue reading the data stream in the hope for an alternative for this description. Only if the result of reading in the data stream is not a complete ODA document even after taking all alternatives into account should the recipient give up decoding the document. This also means that a document that completely misses a particular primary description should not be considered to be in error since that primary description may have been in a part of the document that could be not be decoded.

A special case should be allowed for object class descriptions: If an object class description cannot be decoded (more precisely: if, after completely reading a data stream, there are references to an object class that do not seem to be present in the part of the data stream that could be understood), an error should only be raised if the object class was actually used by an object for which there is no valid alternative description. This special case also obviates the need for alternative descriptions for classes; if an alternative description is to be provided for a class it should be included in all generators for subordinates in a choice together with the primary object class and alternative descriptions should be provided for all objects using the primary class.

G.6 *Preference between several alternatives*

When several alternatives are made available in a document and the recipient can process more than one of them, inability to use the primary description leads to the question which alternative should be used by the recipient. A simple linear priority is provided by the chain created by the “alternate” attribute. This linear priority can also be obtained from the sequence of the alternatives in the interchange stream for the reasons of syntactical fallback given above. To allow for use of the “read until you understand” strategy, [Recommendation T.415/Part 5] also specifies that the alternatives shall be ordered in the data stream by order of their decreasing preference.

IV Addenda to the T.410-Series of Recommendations (1988) on the subject of Styles extension

Within existing § 2.3.5 of Recommendation T.412, *add* the following text following the last sentence of the second paragraph:

Styles may be derived from other styles. The derived style will only specify the attributes and/or attribute values that differ from the style from which it is was derived.

Within existing § 2.3.10 of Recommendation T.412, *add* the following paragraph following the fifth paragraph:

Styles may be present in both the interchange document and the resource-document. If styles in the interchange document and the resource-document have the same identifier then references from the resource-document are to the style in the resource-document, and references from the interchange document are to the style in the interchange document.

Within existing § 5.1.1.2 of Recommendation T.412, *add* the following text following the second paragraph of the subclause:

A style that explicitly specifies all of the appropriate attributes that relate to that style is termed a root style. Any number of additional styles may be derived from a root style. The derived styles will specify only the attributes and/or attribute values that differ from the root style. This provides a means for factoring attributes thus preventing the necessity of copying the same attributes in similar styles. Any number of levels of derived styles may be provided by the first specifying a style derived from the root style and then specifying other styles derived from derived styles.

Within existing § 5.1.1.2 of Recommendation T.412, *replace* the last sentence of the fourth paragraph with the following text:

Precedence rules are specified in §§ 5.1.2.4, 5.1.2.6 and 5.7.12.

Within existing subclause 5.1.1.3 of Recommendation T.412, *add* the following text following the third paragraph of the subclause:

A style that explicitly specifies all of the appropriate attributes that relate to that style is termed a root style. Any number of additional styles may be derived from a root style. The derived styles will specify only the attributes and/or attribute values that differ from the root style. This provides a means for factoring attributes thus preventing the necessity of copying the same attributes in similar styles. Any number of levels of derived styles may be provided by first specifying a style derived from the root style and then specifying other styles derived from derived styles.

Within existing § 5.1.1.3 of Recommendation T.412, *replace* the last sentence of the fifth paragraph with the following text:

Precedence rules are specified in §§ 5.1.2.4 and 5.1.2.6.

Within existing § 5.1.2.4 of Recommendation T.412, *replace* item b) of § 7 with the following text:

If the object description concerned refers to a style then the value of the attribute specified or derived for that style is used (see § 5.1.2.6).

Within existing § 5.1.2.4 of Recommendation T.412, *replace* item d) of § 7 with the following text:

If the object description concerned refers to an object class description which contains a reference to a style then the value of the attribute specified or derived for that style is used (see § 5.1.2.6).

Within existing § 5.1.2.4 of Recommendation T.412, *replace* item f) of § 7 with the following text:

If the object description concerned refers to an object class description which refers to an object class description in the resource-document which contains a reference to a style then the value of the attribute specified or derived for that style is used (see § 5.1.2.6).

Insert the following new subclause immediately after existing § 5.1.2.5 of Recommendation T.412:

5.1.2.6 *Determining values for attributes of styles*

To determine the value of an attribute in a layout style or presentation style, the value is determined by the first of the following rules which is applicable:

a) if an attribute value is specified in the style concerned, then that value is used;

b) if the style concerned is derived from another style, and that style contains a value for the attribute, then that value is used;

c) if the style concerned is derived from another style, and that style is derived from other styles at any number of levels including the root style, then the attribute value determined for the lowest style level is used;

d) if no value is determined by the preceding steps a) to c), then no value is determined for the attribute. (For defaultable attributes see § 5.1.2.4.)

Within existing § 5.6.2 of Recommendation T.412, *add* the following text at the end of the list in the first paragraph:

– derived from (see § 5.10).

Within existing § 5.8.2 of Recommendation T.412, *add* the following text at the end of the list in the first paragraph:

– derived from (see § 5.10).

Insert the following new subclause immediately after § 5.9 of Recommendation T.412:

5.10 *Derived from*

Constituents: Layout styles and presentation styles

Classification: Non-mandatory

Permissible values: A presentation style identifier or a layout style identifier

Definition:

This attribute is used to establish a relationship between a layout or presentation style and another style of the same type. Attributes and their values from the referenced style are used along with the directly specified attributes. Values of directly specified attributes take precedence over values obtained from referenced styles.

Within existing § 5.10 of Recommendation T.415, in "Presentation-Style-Descriptor" and "Layout-Style-Descriptor" *replace* the closing brace (}) with a comma (,) followed by the following line of text:

derived from [x] IMPLICIT Style-Identifier OPTIONAL}

Note – The value of "x" is to be assigned by the editor of Recommendation T.415.

V Extensions on security

V.1 *Changes to Recommendation T.411*

V.2 *Changes to Recommendation T.412*

V.3 *Changes to Recommendation T.414*

V.4 *Changes to Recommendation T.415*

V.1 *Changes to Recommendation T.411*

Amend three definitions in Recommendation T.411 as follows:

3.32 *Constituent*: A set of attributes that is one of the following types: a document profile, an object description, an object class description, a presentation style, a layout style, a content portion description, or a protected part description.

3.47 *Descriptor*: A data structure representing the document profile, an object class description, a layout style, a presentation style, an object description, or a protected part description.

3.53 *Document body*: The part of a document that may include a generic logical and layout structure, specific logical and layout structure, layout and presentation styles, protected parts but excludes the document profile.

Amend the new definitions as follows:

The following definitions shall be merged into § 3.

3.1 *Authenticity:* The property that the claimed data source can be verified to the satisfaction of the recipient.

3.2 *Confidentiality:* The property that information is not made available or disclosed to unauthorized individuals, entities or processes.

Note – This property is limited here to preventing unauthorized semantic knowledge of a document or specified parts of it.

3.3 *Data integrity:* The property that data has not been altered or destroyed in an unauthorized manner.

3.4 *Digital signature:* A form of seal associated with a specified part of a document which provides proof of uniqueness of the identity of the originator (source) who applied the seal; it supports non-repudiation of origin of the sealed (signed) part.

3.5 *Fingerprint:* A short and compact code that may be computed in order to characterize some specified information, with the property that it is not practicable to construct different information which would yield the same output.

3.6 *Integrity:* Used here synonymously with data integrity.

3.7 *Intended recipient:* An intended recipient of a document is a recipient that is expected to receive or have access to the document.

3.8 *Non-repudiation of origin:* The property that an originator can be proved to the satisfaction of a third party to be the source of a document or specified parts of it.

3.9 *Privileged recipient:* A privileged recipient of a document is a recipient that in addition to being an intended recipient, has the right to perform certain security-related operations intended for that particular recipient, such as, to interpret specified enciphered parts of the document, and to perform integrity and authenticity checks on specified parts of the document.

3.10 *Recipient:* A recipient of a document is any object or user receiving or having access to a document.

3.11 *Seal* (noun): Data associated with a specified part of a document by an originator, which a recipient (privileged recipient) may use to verify the integrity and authenticity of the specified part.

3.12 *Seal* (verb): To associate a seal with a specified part of a document.

3.14 *Security domain:* The set of resources subject to a single security policy.

3.15 *Security label:* A marking of a document, which specifies the handling of the document according to the security policy in force.

3.16 *Security policy:* The set of rules that specify the procedures and services required to maintain the intended level of security of a set of resources.

Add a definition to § 3.

3.xx *Protected part:* A constituent consisting of a part of the document that is sealed or enciphered, e.g., a sealed document profile or a pre-enciphered document body part.

Insert the following text before the last paragraph of § 5.1:

It also supports security aspects such as

- protection against unauthorized semantic knowledge of parts of a document;
- detection of discrepancies between the claimed and actual source and content of parts of the document.

Add a new § 5.2.9 to Recommendation T.411:

5.2.9 *Protected parts*

Protected parts enables protection to be provided for parts of a document, offering confidentiality, integrity, authenticity and non-repudiation of origin.

Protected parts cannot provide for the protection of a document as a whole. Treatment of the complete document as an object is outside the scope of this Recommendation.

Parts of a document can be protected before layout in its processable form, or after the layout process in formatted or formatted processable forms.

V.2 *Changes to Recommendation T.412*

Add a bullet item in § 1.1 as follows:

– defines the reference model of the document layout process;

– defines the reference model of the document imaging process;

– defines the reference model for protecting parts of a document;

– defines three document architecture classes;

– defines a notation used for illustrating and describing document structures;

– provides examples of document architecture levels;

– provides examples of document structures;

– provides examples of particular document attributes.

Add to the bulleted list in § 2.3.1

– sealed document profile description;

– enciphered document profile description;

– pre-enciphered document body part description;

– post-enciphered document body part description.

Add a new paragraph between §§ 2.3.6 and 2.3.7

2.3.6.a *Protected part descriptions*

A protected part description is a set of attributes which may be referred to from the document profile.

There are four kinds of protected part descriptions:

– sealed document profile description;

– enciphered document profile description;

– pre-enciphered document body part description;

– post-enciphered document body part description.

Change in § 2.3.12 in the bullet b):

1) constituents representing the generic structure, optionally style constituents, and optionally constituents representing protected parts of the document;

2) constituents representing the specific structure, optionally style constituents, and optionally constituents representing protected parts of the document;

3) constituents representing the generic structure, the specific structure, optionally style constituents, and optionally constituents representing protected parts of the document;

4) constituents representing protected parts of the document.

Add to the bullet list in § 2.3.12:

l) the constituents representing the protected parts of the document consist of sealed document profile descriptions and/or enciphered document profile descriptions and/or pre-enciphered document body part descriptions and/or post-enciphered document body part descriptions.

Replace Figure 2 by:

Figure page entière

Add a new § 2.6

2.6 *Security protection of parts of a document*

This Recommendation distinguishes between the following two sets of security protected parts of a document:

a) parts of a document profile description;

b) parts of the document body consisting of complete object descriptions, object class descriptions, layout styles and presentation styles. A complete composite object description implies its complete substructure including the content portions.

Two concepts of document security exist:

a) Information indicating how the whole document should be handled as a single unit, according to the security policy of the security domain to which the originator belongs. The originator is responsible for making the indication; the actual security handling of the document is outside the scope of both originator and recipient, and outside the scope of the T.410 Series Recommendations.

b) Information to be interchanged between the originator and the recipient on how security aspects of parts of the document should be handled. The handling of this component of security is under the control of the originator and the recipient.

For parts of the document, case b) covers the properties of:

– confidentiality;

– integrity;

– authenticity, including signature and non-repudiation of origin.

It is not the purpose of the T.410 Series Recommendations to specify any particular security scheme or method, but rather to provide the means in the document for a variety of possible security implementations as required by various security policies.

Two cryptographic techniques may be used to provide the security protections in the T.410 Series Recommendations:

– encipherment of clear text to provide confidentiality and possibly integrity of data;

– production of cryptoghraphically derived information to provide integrity and authenticity of data.

There are two phases, a generation phase and an interpretation phase involved with security protection of parts of a document.

The generation phase enciphers or seals parts of the document and creates security information that is added to the document. The protected part descriptions are created in this phase. These descriptions contain the enciphered and the sealed versions of protected parts of the document.

The interpretation phase deciphers enciphered protected parts or validates seals that have been created in the generation phase.

2.6.1 *Intended and privileged recipients*

Two categories of recipients, namely intended recipients and privileged recipients are defined.

An intended recipient of a document is a recipient that is expected to receive or have access to the document.

A privileged recipient of a document is a recipient that, in addition to being an intended recipient, has the right to perform certain security-related operations intended for that particular recipient, such as to interpret certain parts of the document and to perform certain integrity and authenticity checks.

2.6.2 *Protecting parts of the document profile*

Protected parts of a document profile are specified in two sets of document profile descriptions, the set of enciphered document profile descriptions and the set of sealed document profile descriptions.

The sealed document profiles are for integrity, authenticity and non-repudiation of origin, one for each seal of parts of the document profile.

The enciphered document profiles are for confidentiality, one for each encipherment of parts of the document profile.

All information about each sealed or enciphered document profile is found in the document profile.

2.6.3 *Protecting parts of the document body*

The parts of the document body that may be protected are object class descriptions, object descriptions, layout styles and presentation styles.

Confidentiality is based on encipherment and integrity, authenticity and non-repudiation are based on a seal.

For encipherment:

The protection of a composite object description implies that not itself, but all its subordinates including the content portions directly referred to from any of its subordinates are protected.

The protection of a basic object description implies that all the content portions directly referred to from it are protected.

The protection of a basic object description implies that for processable form and form documents, all the content portions directly referred to from it are protected. In a formatted processable form document whereas all content portions directly referred to from a basic object description in one of the structures are protected it may be that only some of the content portions directly referred to from a basic object description in the other structure are protected.

The protection of a composite object class description, layout style or presentation style implies that all of it is protected.

Any component or style that is enciphered shall not be present in the document in its unenciphered form.

For sealing:

The protection of a composite object description implies that itself and all its subordinates including the content portions directly referred to from any of its subordinates are protected.

The protection of a basic object class description implies that the basic component itself and all content portions directly referred to from it are protected.

The protection of a basic object description implies that for processable form and formatted form documents, itself and all the content portions directly referred to from it are protected. In a formatted processable form document whereas all content portions directly referred to from a protected basic object in one of the structures will be protected, it may be that only some of the content portions referred to from a basic object description in the other structure are protected.

The protection of a composite object class description, layout style or presentation style implies that all of it is protected.

For confidentiality enciphered document body parts are defined.

There are two sets of enciphered document body part descriptions: one set for pre-enciphered document body parts and one set for post-enciphered document body parts.

A pre-enciphered document body part description is provided for each encipherment performed before the layout process, and a post-enciphered document body part description is provided for each encipherment performed after the layout process.

All information about each pre- or post-enciphered document body part is found in the document profile.

For integrity, authenticity and non-repudiation of origin seals are provided. The seals and all information about them are found in the document profile.

Add to the bullet list in § 5.1.1:

– protected part attributes.

Change 5.3 – 5.9 in § 5.1.1 to:

5.3 – 5.10.

Add to the bullet list in § 5.1.1.2:

– sealed.

Add to the bullet list in § 5.1.1.3:

– sealed.

Add § 5.1.1.6 as follows:

5.1.1.6 *Protected part attributes*

There are four kinds of descriptions of protected parts of a document, one for sealed information and three for enciphered information. They are:

A sealed document profile description consists of:

– sealed document profile identifier;

– sealed document profile information, which is a document profile, with an identical structure to regular document profile. The differences are that:

  – every attribute is optional;

  – only the attributes that are sealed should be present.

It is also possible to seal absent attributes.

The enciphered document profile description consists of two attributes:

– enciphered document profile identifier;

– enciphered document profile information. The value of this attribute is the result of an encipherment of the confidentially protected part of the document profile. The confidentially protected part of the document profile has the same structure as a regular document profile. The differences are that:

  – every attribute is option;

  – only the attributes that are confidential should be enciphered.

A pre-enciphered document body part description consists of two attributes:

– pre-enciphered document body part identifier;

– pre-enciphered body part information. The value of this attribute is the result of encipherment of the confidential part of the document body applied before the layout process has been performed.

A post-enciphered document body part description consists of two attributes:

– post-enciphered document body part identifier;

– post-enciphered body part information. The value of this attribute is the result of encipherment of the confidential part of the document body applied after the layout process has been performed.

Amend the table in § 5.3.5.5 as follows:

TABLE 1

Defaultable attributes that can be specified in defaut value lists

Object type	Defaultable attributes that can be specified
Document layout root Document logical root Page set	(No attributes can be specified)
Composite or Basic page	Presentation style Content architecture class Content type Dimensions Transparency Colour Page position Medium type Presentation attributes Sealed
Frame	Position Dimensions Border Layout path Permitted categories Transparency Colour Sealed
Block	Presentation style Content architecture class Content type Position Dimensions Border Transparency Colour Presentation attributes Sealed
Composite logical object	Protection Layout style Sealed
Basic logical object	Presentation style Content architecture class Content type Protection Layout style Sealed

Insert a new § 5.3.6

5.3.6 *Security attributes*

5.3.6.1 *Enciphered*

Constituents:

Basic component descriptions and composite object descriptions

Classification:

Non-mandatory.

Structure:

Two parameters, “enciphered subordinates” and “protected part identifier”.

Permissible values:

For the parameter “enciphered subordinates” “none”, “all” or when “partial”, a sequence of one or more non-negative integers. The value “partial” is only applicable to a basic object description in a document of the formatted processable form.

For the parameter “protected part identifier”, which is optional, the identifier of a pre- or a post-enciphered document body part.

Definition:

This attribute specifies if the object descriptions and content portions subordinate to this object component description are enciphered, and where the enciphered parts are found.

When for a composite object description, the parameter “enciphered subordinates” has the value “all”, then all subordinates object descriptions and all content portions directly referred to from any of them are enciphered.

When for a composite object description, this attribute is not specified or the parameter “enciphered subordinates” has the value “none”, then none of its immediate subordinate object descriptions are enciphered.

When for a basic component description, the parameter “enciphered subordinates” has the value “all”, then all content portions directly referred to from it are enciphered.

When for a basic component description, this attribute is not specified or the parameter “enciphered subordinates” has the value “none”, then none of the content portions directly referred to from it are enciphered.

When for a basic object description in a formatted processable form document the parameter “enciphered subordinates” has the value “partial” in the form of a sequence of content portion identifiers, then only the specified content portions are enciphered. Each content portion identifier is represented by its last integer.

When the parameter “enciphered subordinates” has a value other than “none” the parameter “protected part identifier” may specify the identifier of the pre- or post-enciphered document body part where the enciphered part is found.

The layout process shall ignore the subtree of the logical structure for which the parameter “enciphered subordinates” has a value different from “none”.

The imaging process shall ignore the subtree of the . . . . . layout structure for which the parameter “enciphered subordinates” has a value different from “none”.

Note – This attribute specifies the current encipherment status. This attribute shall thus be updated at every encipherment and every decipherment of a document body part.

Paragraph 5.3.6.2, *replace* the text after “Classification:” by:

5.3.6.2 *Sealed*

Constituents:

Component descriptions, and styles.

Classification:

Non-mandatory for object class descriptions;

Non-mandatory for styles;

Defaultable for object descriptions.

Structure:

Two parameters, “sealed status” and “seal identifiers”.

Permissible values:

For the parameter “sealed status”, “no” or “yes”.

For the optional parameter “seal identifiers”, a list of seals in which this component or style is incorporated.

Default value:

Only the parameter “sealed status” has a default value.

The default value is “no”.

Definition:

This attribute specifies if this component description or style is incorporated in a seal for integrity, authenticity or non-repudiation of origin.

The value “no” of the parameter “sealed status” implies that this component description or style is not incorporated in any seal for integrity, authenticity or non-repudiation of origin.

When for a composite object description the parameter “sealed status” has a value “yes”, then it and all its subordinates and all content portions directly referred to from any of them are included in one or more seals.

When for a composite object class description or style the parameter “sealed status” has a value “yes”, then it is included in one or more seals.

When for a basic component description the parameter “sealed status” has a value “yes” then it and all content portions directly referred to from it are included in one or more seals.

The parameter “seal identifier” specifies the seals in which the specified part of the document body is included.

Note – This attribute specifies the current sealing status. This attribute shall thus be updated at every sealing or deletion of a seal of a document body part.

Add to the list in § 5.6.2:

– sealed (see § 5.3.6.2)

Add to the list in § 5.8.2:

– sealed (see § 5.3.6.2)

Insert a new § 5.10.

5.10 *Protected part attributes*

5.10.1 *Protected part identifier*

Constituents:

Protected part descriptions.

Classification:

Mandatory.

Permissible values:

A sequence of two non-negative integers.

The values assigned to the first integer are:

– 6, if the protected part is a sealed document profile description;

– 7, if the protected part is an enciphered document profile description;

– 8, if the protected part is a pre-enciphered document body part description;

– 9, if the protected part is a post-enciphered document body part description;

Representation:

A character string consisting of decimal numerals and space characters. The decimal numerals are in one to one correspondence with the integers constituting the identifier; a space character is used as a separator between successive numerals.

` *Definition:*

This attribute identifies a protected part description uniquely within the context of the document.

5.10.2 *Sealed document profile information*

Constituents:

Sealed document profile description.

Classification:

Non-mandatory.

Permissible values:

Any subset of a document profile.

Representation:

A document profile descriptor, with the additional property of allowing the value “null” for any attribute in the document profile that is not classified as mandatory.

Definition:

This attribute consists of the subset of the attributes of a document profile that is sealed for integrity, authenticity or non-repudiation of origin. A value “null” in an attribute of the sealed document profile indicates that this attribute is sealed as absent.

5.10.3 *Enciphered information*

Constituents:

Enciphered document profile description, pre-enciphered document body part description, post-enciphered document body part description.

Classification:

Non-mandatory.

Permissible values:

Enciphered information.

Representation:

An octet string

Definition:

For an enciphered document profile description, this attribute contains the result from a cryptographic algorithm applied to a confidential part of the document profile.

For a pre-enciphered document body part description, this attribute contains the result from a cryptographic algorithm applied to a sequence of constituents of the document body before the layout process has been performed.

For a post-enciphered document body part description, this attribute contains the result from a cryptographic algorithm applied to a sequence of constituents of the document body after the layout process has been performed.

Add a new paragraph between §§ 7 and 8 (after § 7.3.5):

7b *Reference model for protecting parts of a document*

This paragraph provides a description of an abstract reference model for protecting parts of a document.

The purpose of this clause is to aid the understanding of the semantics of the attributes related to the different aspects of security described in the T.410-Series Recommendations. It does not imply an actual implementation or definition of a standardized process.

The document security processing consists of two phases. One phase enciphers or seals parts of the document and creates security information that is added to the document. The other phase makes use of security information in the document for deciphering a part of the document or checking a seal of a part of the document. These may occur during several phases of document processing, e.g., during the editing process, before the layout process or after the layout process.

The description of the document security model is made in two steps: first an overall model covering the interchange of a document between an originator and a recipient (see § 7b.1); secondly covering the local systems of the originator and the recipient (see § 7b.2).

The local system is here defined as those parts of a system for interchanging documents on which the originator or the recipient have a direct influence, i.e., preparing the document resulting in a valid data stream on the originator's side and after receiving an appropriate data stream on the recipient's side.

The rest of the system consists of parts that are responsible for the actual transfer of the document and those security facilities that implement the security policy of the security domains to which the originator and the recipient belong.

A more detailed description is found in the informative Annex G.

7b.1 *The overall model*

Throughout the following the distinction is drawn between the handling of the complete document by the system and its security facilities and mechanisms, and the handling of specified parts of the document in the possession of the user, an originator or a recipient.

The processes used for the preparation of the data stream belong to the local system of the originator.

The processes used for handling the received data stream belong to the local system of the recipient.

The two local systems are assumed to be able to provide and utilize the security information described here concerning the parts of the document.

The local system may generate information concerning the handling of the complete document by a security facility outside the local system, but this is advisory. This information, an ODA security label, will be interpreted by the security facility in the context of the security policy in force in the security domain to which the originator belongs.

7b.2 *The local system*

The model of the local system describes the security processes involved and their relations to the three processes (editing process, layout process and imaging process) described in the document processing model (see § 2.4).

The local system of the originator prepares the document including the interchange of security information intended either for the recipient or for the security facility of the security domain of the originator (see Figure 2).

Those aspects of the security information intended for the recipient are dealt with by the local system of the recipient and only deal with security protected parts of the document.

Those aspects intended for the security facility of the security domain of the originator not belonging to the local system are specified in the ODA security label of the document profile and this facility will handle the document according to the security policy in force. It can only handle the document as a single unit, i.e., the whole of the document.

The originator can:

- encipher certain parts of the document in order to provide confidentiality, i.e., encipherment;
- provide a seal which allows the recipient to perform checks for:
  - content integrity;
  - origin authenticity;
  - non-repudiation of origin.

Note – Sealing has no influence on the content itself, whereas encipherment will change the content.

The recipient can:

- decipher enciphered parts of the document;
- perform a check on content integrity;
- perform a check on origin authenticity;
- perform a check on non-repudiation of origin.

Security protection can be applied to a document in either processable (PDA), formatted processable (FPDA) or formatted (FDA) form. In other words, the security protection can be performed either before, after or both before and after the layout process. Depending on which form the security protection is applied to, the protection will be different.

Sealing of a document before or after the layout process is called pre- sealed and post-sealed, respectively.

Sealing has no impact on the layout process or the imaging process.

A seal, that has been made on a document can be only checked when the document is in the same form.

Encipherment of a document before or after the layout process has quite different effects. Pre-encipherment is the term used for encipherment of parts of a document before the layout process and post-encipherment when it is performed after the layout process.

Pre-encipherment of a document in processable form will result in a pre-enciphered processable document interchange format.

A layout process will ignore all pre-enciphered parts in a pre-enciphered processable form document. The created layout structure will thus have no knowledge or indication of the existence of any pre-enciphered parts.

A document in processable form or pre-enciphered processable form can serve as input to a layout process. This process will result in one of the forms: formatted processable form, formatted form, pre-enciphered formatted processable form, or pre-enciphered formatted form.

These four forms can be post-enciphered, resulting in the four forms: post-enciphered formatted processable form, post-enciphered formatted form, pre- and post-enciphered formatted processable form, or pre- and post-enciphered formatted form.

The imaging process will ignore all pre- and post-enciphered parts of the document. But since the size and positions of a post-enciphered layout object is specified explicitly in the specific layout structure, the post-enciphered parts within the laid out areas will not be imaged.

The imaging process receiving any of these documents will present them such that:

– all clear text parts will be imaged;

– the pre-enciphered parts will be completely lost in the imaging process;

– the post-enciphered parts will have claimed areas of correct dimensions, but their content is not imaged.

All combinations of protection can be applied to a document, but not all combinations are possible for an individual part of the document.

Add new Annex G to T.412:

ANNEX G

(to Recommendation T.412)

(informative)

Further information on security aspects within a document

G.1 *What can in principle be protected within a document?*

This Recommendation provides two categories of security aspects, namely:

– the incorporation of a security label which provides information of how the originator wants the system to handle the *document as a whole*; and

– the incorporation of security protections of *parts of a document*.

The intended protection that is provided by this Recommendation on the document as a whole is the presence of the ODA security label. Although the security label is not sealed, an integrity protection of it can be achieved by sealing that particular part of the document. It is outside the scope of this Recommendation to provide any other protection mechanism on the document as a whole.

The rest of this paragraph is constrained to the security aspects of *parts of a document* conforming to the T.410-Series Recommendations.

G.1.1 *What does a document contain?*

A document structured according to the T.410-Series Recommendations always contains the document profile. In addition it may contain styles, generic structures and specific structures. A characteristic feature of such a document is that if a structure is present in the document, it is always present in its entirety.

A document that happens to get into the hands of a recipient will thus always contain the document profile. If it is a specific document, it will also always contain a complete document body.

G.1.2 *What can an unauthorized recipient do with a document?*

A recipient can, in principle, do anything that his local system allows with the document. It is here assumed that the recipient has access to a system in which any bit can be accessed, analysed, deleted, changed or added.

In such a case, the recipient can delete and modify any part of the received document as well as adding any part to it. If the generic structure(s) is (are) also interchanged, the recipient can furthermore operate on the document according to the rules specified by the originator. *It is thus not possible to provide any kind of access control to parts of a document.*

G.1.3 *What protection can be made in a document?*

There are two aspects to consider, namely the protection against an unauthorized recipient getting semantic knowledge about a part of a document and the protection against an unauthorized recipient modifying a part of a document.

– What information can be protected?

  If the document contains any enciphered parts, the semantic content can be kept unknown to the recipient as long as the information for decipherment of that content is not known to that recipient.

  A recipient can, however, replace the content and the information that the document once contained in enciphered parts, and can change any clear text by deleting, changing or adding anything to it. Any information in the document profile can be deleted, modified or added.

– What manipulations can be protected against?

  If the unintended recipient, after any of the manipulations discussed under G.1.2, submits the document to the intended recipient, it is natural to ask what protection against such modifications can be made. First we observe that since any bit can be deleted, added or changed by the unintended recipient, *the only mode of protection is in the form of detection of such modifications.*

  Since the authenticity control and/or integrity control of any part of the document is specified in the interchanged document, in our case in the document profile, only changes, not replacements can be protected against. This implies that if the document contained any authenticity, integrity or enciphered part(s), the recipient can only perform a control against a modification if:

  a) either the information about the authenticity, integrity and encipherment is still retained in the document profile; or

  b) the intended recipient knows in advance, e.g. by the security policy, that such a document shall contain such protected parts.

G.1.4 *Summary*

G.1.4.1 *What can be protected within a document?*

It is thus possible to protect a document against the following threats:

a) against semantic knowledge of the content of parts of the document to an unauthorized recipient by means of encipherment;

b) against unauthorized changes to, but not replacements of, parts of a document by means of detection that something, but not what, has been changed (integrity and authenticity controls);

c) against unauthorized replacements of parts or the whole document by means of a pre-agreed security policy between the originator and the intended recipient.

G.1.4.2 *What cannot be protected within a document?*

It is thus not possible to protect a document against the following threats:

a) against detection of replacements of parts or the whole document in an open interchange, i.e. in which no pre-agreed security policies have been made;

b) against deletions, changes or additions to parts of a document by an unauthorized recipient.

A feature that cannot be fulfilled in an open interchange according to the above analysis is the possibility to interchange access control information for parts of a document such that some (intended) recipients may do certain manipulations on that object, e.g. read only, whereas other recipients may have the right to modify it etc. The reason why these security aspects cannot be achieved is that such access cannot be controlled outside a closed environment as seen in § G.1.2 above.

In a closed environment, however, such a protection can be achieved. This can, for example, be done in the form of an application comment, which specifies information that can be interpreted and enforced by all equipment within a particular environment in which the security policy is such that all equipment must be able to interpret and obey that information.

Since such information cannot be enforced outside that environment, it cannot belong to this Recommendation, which deals with security aspects that can be relied upon anywhere in an open environment. It is more like editing instructions (application dependent) that can only be correctly understood and handled in a closed environment. It specifies the intentions rather than the constraints.

G.2 *Security features supported by the T.410-Series Recommendations*

The T.410-Series Recommendations support the interchange of information related to security aspects associated with a document conforming to the T.410-Series Recommendations. This includes the provisions:

- to hide parts of the document from unauthorized persons (confidentiality);
- to check the correctness of parts of the content of the received document (integrity);
- to prove the origin of parts of the received document (authenticity, non-repudiation of origin).

The security aspects of the T.410-Series Recommendations complement the security facilities provided by the OSI and Telematic services.

These security aspects are applicable to parts of a document. In addition, the T.410-Series Recommendations also provide an indication to the system for the handling of the complete document.

The T.410-Series Recommendations do not address the broader aspects of security of systems, including those of the network, or security of workstations and terminals, which are local matters.

G.2.1 *Features provided to an originator*

The T.410-Series Recommendations provide the following security related features for an originator of a document:

- that any intended recipients can interpret the clear text parts of the document, but that only privileged recipients can interpret the clear text and certain additional specified parts of the document (confidentiality);
- that privileged recipients can obtain confirmation that specified parts of the document are intact, i.e. received exactly as originated (integrity);
- that privileged recipients can prove to a third party that specified parts of the document are intact, i.e. exactly as originated (integrity);
- that privileged recipients can obtain confirmation that the claimed originator is the source of specified parts of the document (authenticity);
- that privileged recipients can prove to a third party that the claimed originator is the source of specified parts of the document (authenticity, non-repudiation of origin).

These protection mechanisms are further described in § G.3.

In addition to these requirements on parts of the document, this Recommendation provides a support of the intentions of the originator for the complete document. The security policy of the security domain to which the originator belongs determines what actions to perform on the document, based on the information provided by the originator. This can incorporate topics such as confidentiality, integrity and authenticity for the whole document.

G.2.2 *Features provided to a privileged recipient*

The T.410-Series Recommendations provide the following security related features for a privileged recipient of a document:

- the ability of a privileged recipient to interpret all relevant parts of the document, including those specified parts that are not interpretable by a non-privileged recipient (confidentiality);
- the ability of a privileged recipient to confirm that specified parts of the document are intact, i.e. received exactly as originated (integrity);
- the ability of a privileged recipient to confirm that the claimed originator is the source of specified parts of the document (authenticity);
- the ability of a privileged recipient to prove to a third party that the claimed originator is the source of specified parts of the document, i.e. the purported originator cannot deny being the claimed source of those parts (authenticity, non-repudiation of origin).

G.3 *The kinds of protection mechanisms supported*

G.3.1 *Confidentiality*

Confidentiality in a document concerns the prevention of non-privileged recipients from obtaining semantic knowledge about specified parts.

Confidentiality of parts of a document is provided by the use of encipherment methods controlled by the originator and the privileged recipient.

Confidentiality of the complete document may be indicated (ODA security label) or requested by the originator, but is to be provided by the system, in accordance with the security policy of the domain to which the originator belongs.

G.3.2 *Integrity*

Integrity in a document concerns the provision of information whereby a privileged recipient may verify that the document or specified parts of it have not been changed since the originator requested them to be sealed for that purpose.

The sealing of parts of the document, and the provision of the appropriate seal for use by privileged recipients, is under the control of the originator. The checking of those parts is under the control of the privileged recipient.

Production and checking of integrity information for the complete document may be indicated or requested by the originator, but is to be provided by the system, in accordance with the security policy of the domain to which the originator belongs.

The assurance provided by integrity alone is limited to detection of change; replacement of the complete sealed parts, whether suitably sealed or not, would be undetected.

Some assurance of integrity may also be derived from the valid successful decipherment of parts marked as confidential.

G.3.3 *Authenticity*

Authenticity in a document concerns the provision of information whereby a recipient may verify that the source of the document or specified parts of it, is as claimed.

This property is provided when the integrity seal is such that the privileged recipient can determine the source of the sealed content.

G.3.4 *Non-repudiation of origin*

The property that an originator can prove to a third party to be the source of a document, or specified parts of it, is called non-repudiation of origin.

This property is provided when the integrity and authenticity seal is produced using a digital signature technique such as outlined below (see § G.4.2).

G.4 *Techniques supported by the T.410-Series Recommendations*

G.4.1 *Techniques for confidentiality*

Encipherment:

A document or any part of a document may be enciphered. The encipherment algorithm and information relating to the key(s) for decipherment is specified or indirectly referenced in the document profile. If part of a specific structure in a document is enciphered, this is also marked in the appropriate structure.

The T.410-Series Recommendations support the use of encipherment techniques in which all protected parts of the document can be grouped together in such a way that a privileged recipient only needs to perform a single instance of decipherment, i.e. knowledge of only a single key is required by that recipient. For a set of different privileged recipients, however, the encryption algorithms and keys may be different, so that each specified part of the document can be read exclusively by a certain privileged recipient. But by use of a symmetric key algorithm and exchanging the symmetric key, e.g. by means of an asymmetric key algorithm, several separate privileged recipients can decipher the same parts.

When enciphering styles or object classes, caution should be made that none of these constituents are referred to from any part of the document that is not belonging to the same encipherment.

G.4.2 *Techniques for sealing for content integrity, authenticity and non-repudiation*

Fingerprint:

For sealing, it is convenient to introduce the concept of fingerprint.

A fingerprint is obtained by processing the specified part of the document using a specified algorithm. The main property of the algorithm is that it is computationally infeasible to construct another input to the algorithm resulting in the same output. In general, the fingerprint will be shorter than the information it characterizes (i.e. of the order of bytes rather than kilobytes).

Seal:

A seal for a specified part of the document may be produced by the originator taking the fingerprint for the specified information together with other optional data such as the identity of the originator applying the seal, location, time, etc. and enciphering this using an identified algorithm.

The recipient may decipher the seal, and, depending on the particular qualities of the encipherment algorithm and key, may verify to a known level of assurance the integrity of the information and the authenticity of the claimed source, as follows:

– Integrity of the specified information may be checked by, firstly, re-running the fingerprint process and comparing the result with the associated fingerprint received in the document, and secondly, that the same fingerprint being used in calculating the seal.

– Authenticity of origin of the specified information may be checked as for integrity, and that the seal is composed such that the recipient can, to his own satisfaction, verify the originator.

– Non-repudiation of origin of the specified information, as a special case of origin authenticity, may be provided if a digital signature process is used for sealing; in this case, the integrity of the information and the authenticity of the source may be proven to a third party, and the source who applied the seal cannot deny responsibility for it; the particular quality of the digital signature is most readily provided by the use of an asymmetric key crypto-system, where the secret ("private") key is allocated to a single originator by a trusted key certification authority, and the corresponding ("public") key may be made available by an authority to authorized recipients.

G.5 *Further details on the reference model for protecting parts of a document*

G.5.1 *The overall model*

This clause provides a more detailed description of the model than found in § 7b.1.

Figure G.1/T.412 illustrates the processes involved in the interchange of a document between the local system of the originator and the transfer system.

The local system of the originator exports a document with the data stream (A). This document may contain protected parts. It may also contain an ODA security label.

In the case where the originator belongs to a security domain in which the security policy requires documents to be associated with a security label, then a security label envelope enclosing the document will be generated. This generation is performed by a security facility immediately outside the local system. When determining the value of the security label, the ODA security label can be taken into account.

The process applied to the exported data stream is completely directed by the security policy of the security domain to which the originator belongs. The responsibility of the process is to take appropriate actions on the whole document, e.g. encipher it.

According to the security policy in force, an operation F on the data stream will take place. If no action takes place, F is the identity operator I with the result that $F(A) = I(A) = A$.

The transfer system does not need to have any understanding of ODA. It is responsible for the physical interchange of the document. Typically this may be achieved by MHS/MOTIS, FTAM, a floppy disk, etc.

Figure G.2/T.412 illustrates the processes involved in the interchange of a document between the transfer system and the local system of the recipient.

The data stream delivered by the transfer system is identical to that entering it, i.e. F(A) in Figures G.1/T.412 and G.2/T.412.

The process applied to the received data stream F(A) serves the purpose of transforming the data stream back to the format of A. This implies finding the inverse function F^{-1} and also to perform this operation on the data stream.

Depending on the security policy it may provide a new security label envelope. It finally also provides the document without an envelope to, e.g. an editor.

The resulting data stream from this process is the one imported by the local system of the recipient.

G.5.2 *The local system*

This clause provides a more detailed description of the local system than found in § 7b.2.

If an ODA security label shall be specified in the document, the security policy of the security domain of the originator specifies:

a) the ODA security label to be associated with the document according to the content of the document;

b) how the security facility shall handle a document according to its ODA security label.

In our reference model of the local system, the security attributes other than the ODA security label are processed by a Security Handler in the local system.

The originator of a document will use the security handler in a different way than the recipient of the document.

Figure G.3/T.412 illustrates the aspects of handling confidentiality in the local system. The editing process, layout process and imaging process are illustrated as in Recommendation T.411. The Security Handlers for enciphering parts of a document are marked by rectangles with a bold solid border and those for deciphering parts of a document with double borders.

It should be pointed out that a document that bypasses the Security Handler on the recipient side will still be interpretable as a normal document with the exception that any enciphered parts will not be imaged and that no check on seals has been made.

The security handler can be applied to a document in either processable, formatted processable or formatted form. In other words, the security processing can be performed either before, after or both before and after the layout process. Depending on which form the Security Handler is applied to, the protection will be different.

A recipient may check for integrity and origin authenticity by means of a seal. This seal, which is provided by the originator, is composed of a fingerprint of the parts of the document to be validated together with optional additional information, such as time, place, name, etc. It is then enciphered such that the authenticity can be verified to the satisfaction of the recipient.

If the encipherment method used in the seal is performed by means of some “public key” method, it provides a digital signature. A digital signature may be checked by any recipient having access to the public key. In cases based on a symmetric crypto system, the check may only be done by privileged recipients in possession of appropriate key and algorithm information. This latter method is less powerful in the sense that it cannot be used to convince a third party of the authenticity, or to protect against forgery by the recipient.

Since the Security Handler did not change the content itself when evaluating a seal, it is quite possible to perform this processing on a document both before and after the layout process without constraints. This is not true for encipherment.

A privileged recipient receiving a document in a pre-enciphered form can perform a pre-decipherment before the layout process acts on the document. If this is not done, e.g. by an intended but not privileged recipient, the resulting document interchange format pre-enciphered formatted processable or pre-enciphered formatted form will be presented by an imaging process with invisible enciphered parts. Not even empty areas of the enciphered parts will be visible.

In the pre-enciphered formatted form all information about the enciphered data is lost and it is not possible to recover the information. Thus, this document interchange format is in fact a formatted form document, but smaller than that originally prepared by the originator.

All forms can be used as interchange formats. These are illustrated in Figure G.3/T.412. The figure also illustrates how the different interchanged formats should be handled in order to retrieve the full information of the document as well as what will be the result of an imaging process handling a non-deciphered version of the document.

G.6 *Document application profiles*

This Recommendation provides a great variety of ways to protect parts of a document. As for other properties of this Recommendation it is the task of the document application profiles to specify for each requirement the optimum solution.

Figure G.1 = 16 cm

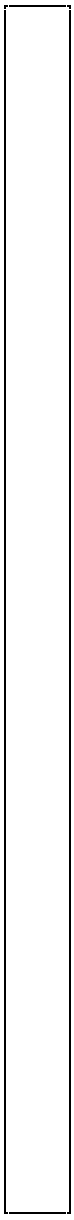


Figure G.2 = 16 cm

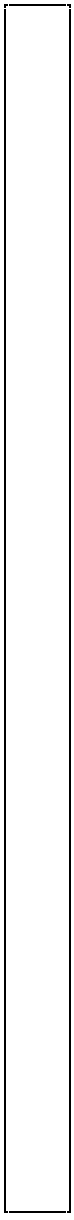


Figure G.3

page entière

V.3 *Changes to Recommendation to T.414*

Insert the following clauses between § 5.2.6 and 5.2.7

5.2.6a *Sealed document profiles*

This attribute is used if and only if the document contains any sealed document profile descriptions.

The value of this attribute (if specified) is “present”.

5.2.6b *Enciphered document profiles*

This attribute is used if and only if the document contains any enciphered document profile descriptions.

This value of this attribute (if specified) is “present”.

5.2.6c *Pre-enciphered document body parts*

This attribute is used if and only if the document contains any pre-enciphered document body part descriptions.

The value of this attribute (if specified) is “present”.

5.2.6d *Post-enciphered document body parts*

This attribute is used if and only if the document contains any post-enciphered document body part descriptions.

The value of this attribute (if specified) is “present”.

Add a new § 5.5

5.5 *Security attributes*

These attributes provide information about those parts of the document that are protected in a secure manner, i.e. provide confidentiality, integrity, authenticity or non-repudiation.

The first attribute is concerned with an indication to the system on how to treat the document as a whole.

All the other attributes in this paragraph are concerned with protected parts of the document.

The attributes described in §§ 5.5.2 to 5.5.4 are concerned with integrity, authenticity and non-repudiation.

The attributes described in §§ 5.5.5 to 5.5.7 are concerned with confidentiality.

When any of these attributes are specified, they must be present in the regular document profile preceding the document body.

5.5.1 *ODA security label*

This attribute specifies the ODA security label associated with this document.

It has two parameters:

– ODA label text; the value consists of a string of characters from the document profile character set;

– ODA label data; the value consists of an octet string.

5.5.2 *Sealed document profiles*

This attribute specifies information associated with each sealed document profile; where to find the sealed document profile, the privileged recipients associated with it and the information needed to check its seal.

It consists of a sequence of parameters, one for each sealed document profile. The sequence of parameters are:

- "sealed document profile identifier", which uniquely identifies the constituent of the sealed document profile;
- "privileged recipients", which consist of a list of names of privileged recipients associated with this sealed document profile;
- "document profile seal", which consists of three sub-parameters:
  - "seal method", which identifies the seal algorithm used;
  - "sealed information", which identifies what has been sealed;
  - "seal", which is the seal.

5.5.3 *Pre-sealed document body parts*

This attribute specifies information associated with each pre-sealed document body part; which body parts have been sealed, the privileged recipients associated with it and the information needed to check its seal.

It consists of a sequence of parameters, one for each pre-sealed document body part. The sequence of parameters are:

- "seal identifier", which identifies the seal;
- "privileged recipients", which consists of a list of the names of privileged recipients associated with this pre-sealed document body part;
- "sealed constituents", which consist of a sequence of the sealed constituents;
- "document body part seal", which consists of three sub-parameters:
  - "seal method", which identifies the seal algorithm used;
  - "sealed information", which identifies what has been sealed;
  - "seal", which is the seal.

5.5.4 *Post-sealed document body parts*

This attribute specifies information associated with each pre-sealed document body part; which body parts have been sealed, the privileged recipients associated with it and the information needed to check its seal.

It consists of a sequence of parameters, one for each post-sealed document body part. The sequence of parameters are:

- "seal identifier", which identifies the seal;
- "privileged recipients", which consists of a list of the names of privileged recipients associated with this post-sealed document body part;
- "sealed constituents", which consists of a sequence of the sealed constituents;
- "document body part seal", which consists of three sub-parameters:
  - "seal method", which identifies the seal algorithm used;
  - "sealed information", which identifies what has been sealed;
  - "seal", which is the seal.

5.5.5 *Enciphered document profiles*

This attribute specifies information associated with each enciphered document profile; where to find the enciphered document profile, the privileged recipients associated with it and the information needed to decipher it.

It consists of a sequence of parameters, one for each enciphered document profile. The sequence parameters are:

- "enciphered part identifier", which identifies the enciphered document profile;
- "privileged recipient information", which consists of three sub-parameters:

- "privileged recipients", which consists of a list of the names of privileged recipients associated with this enciphered document profile;
- "method information", which identifies the encipherment algorithm used;
- "key information", which provides information for the privileged recipient to deduce the key needed to decipher the enciphered document profile; this parameter has two sub-sub-parameters:
  - "key method", which identifies the method how to deduce the key;
  - "additional information", which in combination with the previous sub-sub-parameter provides the privileged recipient with information on how to deduce the key needed for the deciphering.

5.5.6 *Pre-enciphered document body parts*

This attribute specifies information associated with each pre-enciphered document body part; where to find the enciphered document body part, the privileged recipients associated with it and the information needed to decipher it.

It consists of a sequence of parameters, one for each pre-enciphered document body part. The sequence of parameters are:

- "enciphered part identifier", which identifies the pre-enciphered document body part;
- "privileged recipient information", which consists of three sub-parameters:
  - "privileged recipients", which consists of a list of the names of privileged recipients associated with this pre-enciphered document body part;
  - "method information", which identifies the encipherment algorithm used;
  - "key information", which provides information for the privileged recipient to deduce the key needed to decipher the pre-enciphered document body part, this parameter has two sub-sub-parameters:
    - "key method", which identifies the method how to deduce the key;
    - "additional information", which in combination with the previous sub-sub-parameter provides the privileged recipient with information on how to deduce the key needed for the deciphering.

5.5.7 *Post-enciphered document body parts*

This attribute specifies information associated with each post-enciphered document body part; where to find the enciphered document body part, the privileged recipients associated with it and the information needed to decipher it.

It consists of a sequence of parameters, one for each post-enciphered document body part. The sequence of parameters are:

- "enciphered part identifier", which identifies the post-enciphered document body part;
- "privileged recipient information", which consists of three sub-parameters:
  - "privileged recipients", which consists of a list of the names of privileged recipients associated with this post-enciphered document body part;
  - "method information", which identifies the encipherment algorithm used;
  - "key information", which provides information for the privileged recipient to deduce the key needed to decipher the post-enciphered document body part; this parameter has two sub-sub-parameters:
    - "key method", which identifies the method how to deduce the key;
    - "additional information", which in combination with the previous sub-sub-parameter provides the privileged recipient with information on how to deduce the key needed for the deciphering.

V.4 *Changes to Recommendation T.415*

Add four items after "– text unit" in § 5.1

– sealed document profile descriptor;
– enciphered document profile descriptor;
– pre-enciphered document body part descriptor;
– post-enciphered document body part descriptor.

Add four items after "– text unit" in § 5.2

– sealed document profile descriptor;
– enciphered document profile descriptor;
– pre-enciphered document body part descriptor;
– post-enciphered document body part descriptor.

Add four items after item i) in § 5.2

j) sealed document profile descriptors;
k) enciphered document profile descriptors;
l) pre-enciphered document body part descriptors;
m) post-enciphered document body part descriptors.

Add three items after "– text unit" in § 5.3

– sealed document profile descriptors;
– enciphered document profile descriptors;
– post-enciphered document body part descriptors.

Add three items after item d) in § 5.3

e) sealed document profile descriptors;
f) enciphered document profile descriptors;
g) post-enciphered document body part descriptors.

Change in first paragraph of § 5.4

"or layout style descriptor"

to

", layout style descriptor, sealed document profile descriptor, enciphered document profile descriptor, pre-enciphered document body part descriptor or post-enciphered document body part descriptor"

Change in second paragraph of § 5.4

"and each object"

to

", each object and each protected part"

Add a new paragraph between §§ 5.4 and 5.5.

5.4a *ASN.1 encoding and cryptographic techniques*

5.4a.1 *Enciphered information*

The parts of the document body or the parts of the document profile which are the output of an encipherment process will form a new constituent of the document. It consists of an identifier and the enciphered information. The latter is of the ASN.1 "OCTET STRING" type, the value of which will remain unchanged in any transfer.

5.4a.2 *Sealed information*

The ODA security attributes and ODA document parts are defined in ASN.1. To ensure a unique encoding of ASN.1, the ASN.1 Distinguished Encoding Rules are used. These rules are defined in Annex E and information on

how they can be used is found in Annex F. The ASN.1 Distinguished Encoding Rules specify a set of restrictions on the ASN.1 Basic Encoding Rules, which provide a unique mapping between ASN.1 and its representation. This is required from a cryptographical point of view.

The parts of the document profile and the parts of the document body subject to sealing will remain unchanged after the sealing process. The ASN.1 Distinguished Encoding Rules will assure that the same encoding of the information can be established by the recipient as that used by the originator when sealing. This is necessary in order to obtain identical fingerprints of the information, the means by which one associates the content with the seal.

The seal is composed of a set of data. Three basic steps are performed to generate this seal:

a) The chosen information (encoded according to the distinguished encoded rules) is input to a hashing process which generates a fingerprint. The encoded form of the fingerprint being an "OCTET STRING".

b) The fingerprint together with additional optional information is called "Sealed-Information". The optional parameters are the date and time of day, in accordance with ISO 8601, the name and the location of the creator of the seal. This is (again encoded according to the ASN.1 Distinguished Encoding Rules) input to a cryptographic process which generates the seal. The encoded form of the seal being an "OCTET STRING".

c) Provide information of the seal method such that the seal can be checked. This is specified in the type "Seal Method" and consists of information of both the generation of the fingerprint as well as information of how to decipher the seal.

The order of the constituents is the same as the one specified by the interchange format class.

When the order of the constituents is not completely specified by the interchange format class, the following rules apply:

– object classes are to be sealed in the same order as they are specified in the parameter "sealed constituents";

– for interchange format class A, the common content portions are to be sealed in the same order as the corresponding object classes;

– presentation styles are to be sealed in the same order as they are specified in the parameter "sealed constituents";

– layout styles are to be sealed in the same order as they are specified in the parameter "sealed constituents".

Add after "FROM Text-Units" in IMPORTS in § 5.5

Sealed-Doc-Prof-Descriptor,

Enciphered-Doc-Prof-Descriptor,

Preenciphered-Bodypart-Descriptor,

Postenciphered-Bodypart-Descriptor

FROM Protected-Part-Descriptors; **--** See § 5.13 **--**

"(Editing instruction: *replace* ";" by "," after "Text-Units".)"

Add after the line "layout-style" in § 5.5

sealed-doc-prof-descriptor **[9] IMPLICIT Sealed-Doc-Prof-Descriptor,**

enciphered-doc-prof-descriptor **[10] IMPLICIT Enciphered-Doc-Prof-Descriptor,**

preenciphered-bodypart-descriptor **[11] IMPLICIT Preenciphered-Bodypart-Descriptor,**

postenciphered-bodypart-descriptor **[12] IMPLICIT Postenciphered-Bodypart-Descriptor }**

(Editing instruction: Change the "}" to "," after "Layout-Style-Descriptor".)

Insert before ";" in EXPORTS in § 5.6:

, Personal-Name

Insert after "Object-or-Class-Identifier" in IMPORTS in § 5.6

Protected-Part-Identifier, Style-Identifier

Insert after data item "layout-styles" in the Document-Profile-Descriptor of § 5.6

```
sealed-profiles                 [12] IMPLICIT NumericString OPTIONAL,
enciphered-profiles             [13] IMPLICIT NumericString OPTIONAL,
preenciphered-bodyparts         [14] IMPLICIT NumericString OPTIONAL,
postenciphered-bodyparts        [15] IMPLICIT NumericString OPTIONAL,
```

Insert after the line "document-management-attributes" in the Document-Profile-Descriptor of § 5.6

```
document-security-attributes    [16] IMPLICIT Document-Security-Attributes
                                     OPTIONAL }
```

(Editing instruction: change the "}" to "," after "Document-Management-Attributes OPTIONAL".)

Insert before "END" in § 5.6

```
Document-Security-Attributes    :: = SET {
        oda-security-label      [0] IMPLICIT Oda-Security-Label OPTIONAL,
        sealed-doc-profiles     [1] IMPLICIT Sealed-Doc-Profiles OPTIONAL,
        presealed-doc-bodyparts [2] IMPLICIT Sealed-Doc-Bodyparts OPTIONAL,
        postsealed-doc-bodyparts [3] IMPLICIT Sealed-Doc-Bodyparts OPTIONAL,
        enciphered-doc-profiles [4] IMPLICIT Protected-Doc-Parts OPTIONAL,
        preenciphered-doc-bodyparts [5] IMPLICIT Protected-Doc-Parts OPTIONAL,
        postenciphered-doc-bodyparts [6] IMPLICIT Protected-Doc-Parts OPTIONAL
        }

Oda-Security-Label              :: = SEQUENCE {
        oda-label-text          [0] IMPLICIT Character-Data OPTIONAL,
        oda-label-data          [1] IMPLICIT OCTET STRING OPTIONAL
        }

Seal-Data                       :: = SEQUENCE {
        seal-method             [0] IMPLICIT Seal-Method OPTIONAL,
        sealed-information      [1] IMPLICIT Sealed-Information OPTIONAL,
        seal                    [2] IMPLICIT OCTET STRING
        }

Seal-Method                     :: = SEQUENCE {
        fingerprint-method      [0] IMPLICIT Method-Information OPTIONAL,
        fingerprint-key-information [1] IMPLICIT Key-Information OPTIONAL,
        sealing-method          [2] IMPLICIT Method-Information OPTIONAL,
        sealing-key-information [3] IMPLICIT Key-Information OPTIONAL
        }

Sealed-Information              :: = SEQUENCE {
        fingerprint             [0] IMPLICIT OCTET STRING OPTIONAL,
        time                    [1] IMPLICIT Date-and-Time OPTIONAL,
        sealing-orig-id         [2] IMPLICIT Personal-Name OPTIONAL,
        location                [3] IMPLICIT Location OPTIONAL
        }

Method-Information              :: = SEQUENCE {
        unique-method-info      [0] IMPLICIT OBJECT IDENTIFIER OPTIONAL,
        descriptive-method-info [1] IMPLICIT Character-Data OPTIONAL
        }
```

```
Key-Information                 :: = SEQUENCE {
        method-information      [0] IMPLICIT Method-Information OPTIONAL,
        additional-information  [1] IMPLICIT Additional-Information OPTIONAL
        }

Additional-Information          :: = SEQUENCE {
        descriptive-information [0] IMPLICIT Character-Data OPTIONAL,
        octet-string            [1] IMPLICIT OCTET STRING OPTIONAL
        }

Location                        :: = SEQUENCE {
        unique-location         [0] IMPLICIT OBJECT IDENTIFIER OPTIONAL,
        descriptive-location    [1] IMPLICIT Character-Data OPTIONAL
        }

Sealed-Doc-Profiles             :: = SET OF SEQUENCE {
        sealed-doc-prof-descriptor-id [0] IMPLICIT Protected-Part-Identifier,
        privileged-recipients   [1] IMPLICIT SET OF Personal-Name OPTIONAL,
        doc-prof-seal           [2] IMPLICIT Seal-Data
        }

Sealed-Doc-Bodyparts            :: = SET OF SEQUENCE {
        seal-id                 [0] IMPLICIT INTEGER,
        sealed-constituents     [1] IMPLICIT Sealed-Constituents,
        privileged-recipients   [2] IMPLICIT SET OF Personal-Name OPTIONAL,
        doc-bodypart-seal       [3] IMPLICIT Seal-Data
        }

Sealed-Constituents             :: = SEQUENCE {
        object-class-identifiers [0] IMPLICIT SEQUENCE OF Object-or-Class-Identifier
                                                OPTIONAL,
        presentation-style-identifiers [1] IMPLICIT SEQUENCE OF Style-Identifier
                                                OPTIONAL,
        layout-style-identifiers [2] IMPLICIT SEQUENCE OF Style-Identifier
                                                OPTIONAL,
        object-identifiers      [3] IMPLICIT SEQUENCE OF Object-or-Class-Identifier
                                                OPTIONAL

        }

Protected-Doc-Parts             ::= SET OF SEQUENCE {
        protected-doc-part-id   [0] IMPLICIT Protected-Part-Identifier,
        priv-recipients-info    [1] IMPLICIT SET OF Priv-Recipients-Info
        }

Priv-Recipients-Info            :: = SEQUENCE {
        privileged-recipients   [0] IMPLICIT SET OF Personal-Name OPTIONAL,
        encipherment-method-info [1] IMPLICIT Method-Information OPTIONAL,
        encipherment-key-info   [2] IMPLICIT Key-Information OPTIONAL
        }
```

Add after “style-identifier” in EXPORTS in § 5.7

, Protected-Part-Identifier

Insert before the data item “Layout-Category-Name” in § 5.7

```
Protected-Part-Identifier       :: = [APPLICATION 7] IMPLICIT PrintableString
```

-- only digits and space are used in the present version of this [R|IS]; other characters are reserved for extensions; a “null” value is represented by an empty string --

Insert after “Border” in EXPORTS of § 5.8

Enciphered, Sealed.

Change in IMPORTS of § 5.8

IMPORTS Object-or-Class-Identifier, Style-Identifier,

to

IMPORTS Object-or-Class-Identifier, Style-Identifier, Protected-Part-Identifier, -- see § 5.6 --

Insert before the data item “Layout-Object-Descriptor” in § 5.8

```
Enciphered                      :: = SEQUENCE {
    enciphered-subordinates     CHOICE {
    none-all                    [0] IMPLICIT INTEGER { none(0), all(1) },
    partial                     [1] IMPLICIT SEQUENCE OF Numeric String},
    protected-part-id           [2] IMPLICIT Protected-Part-Identifier OPTIONAL
    }

Sealed                          :: = SEQUENCE {
    sealed-status               [0] IMPLICIT INTEGER { no(0), yes(1) },
    seal-ids                    [1] IMPLICIT SET OF INTEGER OPTIONAL
    }
```

Add to both Layout-Object-Descriptor-body and to Layout-Class-Descriptor-Body in § 5.8

```
    enciphered                  [27] IMPLICIT Enciphered OPTIONAL,
    sealed                      [28] IMPLICIT Sealed OPTIONAL}
```

(Editing instruction: Change the “}” to “,” after the last “OPTIONAL” of both the “Layout-Object-Descriptor-Body” and the “Layout-Class-Descriptor-Body”)

Change in IMPORTS of § 5.9

IMPORTS Object-or-Class-Identifier, Style-Identifier,

to

IMPORTS Personal-Name
FROM Document-Profile-Descriptor
Object-or-Class-Identifier, Style-Identifier, -- see § 5.6 --

Insert after “Binding-Pair” in IMPORTS in § 5.9

Enciphered, Sealed.

Add to both Logical-Object-Descriptor-Body and to Logical-Class-Descriptor-Body in § 5.9

```
    enciphered                  [27] IMPLICIT Enciphered OPTIONAL,
    sealed                      [28] IMPLICIT Sealed OPTIONAL}
```

(Editing instruction; Change the “}” to “,” after the last “OPTIONAL” of both the “Logical-Object-Descriptor-Body” and the “Logical-Class-Descriptor-Body”)

Insert before “Object-or-Class-Identifier” in IMPORTS of § 5.10

Sealed

Add to both Presentation-Style-Descriptor and to Layout-Style-Descriptor in § 5.10

```
    sealed                      [6] IMPLICIT Sealed OPTIONAL}
```

(Editing instruction: Change the "}" to "," after the last "OPTIONAL" of both the "Presentation-Style-Descriptor" and the "Layout-Style-Descriptor")

Change § 5.11 of Recommendation T.415 as follows:

5.11 *Default value lists*

```
Default-Value-Lists { 2 8 1 5 11 }

DEFINITIONS        :: = BEGIN

EXPORTS Default-Value-Lists-Logical, Default-Value-Lists-Layout;

IMPORTS Style-Identifier, Layout-Category-Name
        FROM Identifiers-and-Expressions                    -- see § 5.7 --
        Measure-Pair, One-Of-Four-Angles, Medium-Type,
        Dimension-Pair, Transparency, Colour, Border, sealed
        FROM Layout-Descriptors                             -- see § 5.8 --
        Protection FROM Logical-Descriptors                 -- see § 5.9 --
        Presentation-Attributes
        FROM Style-Descriptors;                             -- see § 5.10 --

Default-Value-Lists-Layout          :: = SET {
        page-attributes             [2] IMPLICIT Page-Attributes OPTIONAL,
        frame-attributes            [3] IMPLICIT Frame-Attributes OPTIONAL,
        block-attributes            [4] IMPLICIT Block-Attributes OPTIONAL
        }

Default-Value-Lists-Logical         :: = SET {
        composite-logical-attributes [5] IMPLICIT Composite-Logical-Attributes OPTIONAL,
        basic-logical-attributes    [6] IMPLICIT Basic-Logical-Attributes OPTIONAL
        }

Page-Attributes                     :: = SET {
        dimensions                  <Attribute OPTIONAL,
        transparency                <Attribute OPTIONAL,
        presentation-attributes     <Attribute OPTIONAL,
        page-position               <Attribute OPTIONAL,
        medium-type                 <Attribute OPTIONAL,
        presentation-style          <Attribute OPTIONAL,
        colour                      <Attribute OPTIONAL,
        sealed                      <Attribute OPTIONAL
        }

Frame-Attributes                    :: = SET {
        position                    <Attribute OPTIONAL,
        dimensions                  <Attribute OPTIONAL,
        transparency                <Attribute OPTIONAL,
        layout-path                 <Attribute OPTIONAL,
        permitted-categories        <Attribute OPTIONAL,
        colour                      <Attribute OPTIONAL,
        border                      <Attribute OPTIONAL,
        sealed                      <Attribute OPTIONAL
        }

Block-Attributes                    :: = SET {
        position                    <Attribute OPTIONAL,
        dimensions                  <Attribute OPTIONAL,
        transparency                <Attribute OPTIONAL,
        presentation-attributes     <Attribute OPTIONAL,
        presentation-style          <Attribute OPTIONAL,
        colour                      <Attribute OPTIONAL,
        border                      <Attribute OPTIONAL,
        sealed                      <Attribute OPTIONAL
        }
```

```
Composite-Logical-Attributes        :: = SET {
        protection                  <Attribute OPTIONAL,
        layout-style                <Attribute OPTIONAL,
        sealed                      <Attribute OPTIONAL
        }

Basic-Logical-Attributes            :: = SET {
        presentation-attributes     <Attribute OPTIONAL,
```

-- only for use for the attribute content-architecture-class the content architecture specific attributes can only be referenced by use of presentation style --

```
        protection                  <Attribute OPTIONAL,
        presentation-style          <Attribute OPTIONAL,
        layout-style                <Attribute OPTIONAL,
        sealed                      <Attribute OPTIONAL
        }

Attribute                           :: = CHOICE {
        position                     [0] IMPLICIT Measure-Pair,
        dimensions                   [1] IMPLICIT Dimension-Pair,
        transparency                 [2] IMPLICIT Transparency,
        presentation-attributes      [3] IMPLICIT Presentation-Attributes,
        layout-path                  [4] IMPLICIT One-Of-Four-Angles,
        page-position                [5] IMPLICIT Measure-Pair,
        medium-type                  [6] IMPLICIT Medium-Type,
        permitted-categories         [7] IMPLICIT SET OF Layout-Category-Name,
        protection                   [8] IMPLICIT Protection,
        presentation-style           [9] IMPLICIT Style-Identifier,
        layout-style                [10] IMPLICIT Style-Identifier,
        colour                      [11] IMPLICIT Colour,
        border                      [12] IMPLICIT Border,
        sealed                      [13] IMPLICIT Sealed
        }

        END
```

5.13 *Protected part descriptors*

```
Protected-Part-Descriptors { 2 8 1 5 13 }

DEFINITIONS        :: = BEGIN

EXPORTS Sealed-Doc-Prof-Descriptor,
        Enciphered-Doc-Prof-Descriptor,
        Preenciphered-Bodypart-Descriptor,
        Postenciphered-Bodypart-Descriptor;

IMPORTS Document-Profile-Descriptor
        FROM Document-Profile-Descriptor                    -- see § 5.6 --
        Protected-Part-Identifier
        FROM Identifiers-and-Expressions                    -- see § 5.7 --

Sealed-Doc-Prof-Descriptor          :: = SEQUENCE {
        sealed-doc-prof-identifier  Protected-Part-Identifier,
        sealed-doc-prof-information Document-Profile-Descriptor
        }

Enciphered-Doc-Prof-Descriptor      :: = SEQUENCE {
        enciphered-doc-prof-identifier  Protected-Part-Identifier,
        enciphered-doc-prof-information Enciphered-Information
        }
```

```
Preenciphered-Bodypart-Descriptor        :: = SEQUENCE {
        preenciphered-bodypart-identifier    Protected-Part-Identifier,
        preenciphered-bodypart-info          Enciphered-Information
        }

Postenciphered-Bodypart-Descriptor       :: = SEQUENCE {
        postenciphered-bodypart-identifier   Protected-Part-Identifier,
        postenciphered-bodypart-info         Enciphered-Information
        }

Enciphered-Information                   :: = OCTET STRING

        END
```

Insert in Annex B

APPLICATION 7	**Protected-Part-Identifier**	§ 5.7

Insert in Annex C

{ 2 8 1 5 13 }	**Identifies Module Protected-Part-Descriptors**	§ 5.13

[Add two new Annexes, Annex E and Annex F|Insert two new Annexes before Annex E and rename the old Annex E and Annex F to Annex G and Annex H, respectively]

ANNEX G

(to Recommendation T.415)

(normative)

The ASN.1 Distinguished Encoding Rules

G.1 *Overview*

The following text specifies the Distinguished Encoding Rules as additional constraints on encoders of ASN.1 types. The encodings may be used for transmission when either the ASN.1 Basic Encoding Rules or the ASN.1 Distinguished Encoding Rules have been agreed as specifying the transfer syntax. They may only be assumed by a receiver if the ASN.1 Distinguished Encoding Rules has been agreed as specifying the transfer syntax.

The model is based on taking values identified at the abstract syntax level and removing all options from the Basic Encoding Rules, thus providing a one-to-one mapping between abstract values and a bitstring. It is important to recognize that when applying this model we are talking about abstract syntax values, not about local representations. In particular, the abstract syntax value of a character in a character string identifies both the character and the register entry it is notionally taken from, and the abstract syntax value for a real is a (mathematical) real number (of arbitrary precision and size), capable of a finite representation using base 2 or using base 10.

Where the following text makes no mention of a type, the Basic Encoding Rules provide no options, and shall be applied unchanged.

G.2 *String types*

Where the type definition specifies the zero trailing bits in a bitstring are not significant, trailing zero bits shall be eliminated before applying the rules below. Where the type definition specifies that zero trailing bits in a bitstring are significant, they shall be included in the encoding.

Octetstrings and bitstrings (and types derived from them by tagging or subtyping) shall be encoded with a primitive encoding if they contain less than 1000 octets or 8000 bits respectively, and as a constructed encoding if they contain more octets or bits than this. The constructed encoding shall use fragments of 1000 octets or 8000 bits for all fragments except the last, and each fragment shall be encoded with a primitive encoding.

Each unused bit in the final octet of the encoding of a bitstring value shall be set to zero.

G.3 *Length forms*

The indefinite length shall be used for all constructed encodings, and the definite length form shall be used for all primitive encodings.

Where the definite length form is used, the short form shall be used in all cases where the contents octets are less than or equal to 127 octets.

In the long form, the minimum number of octets shall be used to encode the length.

G.4 *Character strings*

For a character string type, the value is defined as a character from a specified register entry in the International Register of Character Sets. (For PrintableString and NumericString, the encoding is separately specified in the Basic Encoding Rules, and this shall be used.)

Where the Distinguished Encoding Rules are being used by a sender, the choice shall be consistently made for the application of the Distinguished Encoding Rules and the choice made in determining the encoding for any subsequent transmission of the character string. Where the Distinguished Encoding Rules are being used to encode a value which has been received, the transmission will have identified the register entry for each character.

The encoding shall generate escape sequences to designate and invoke a new register entry only when the register entry for the character is different from that currently designated as G0, C0 or C1. All designations and invocations shall be into the G0 set or the C0 set.

G.5 *Components of sets and sequences*

If the value of a component of a set or sequence which is marked DEFAULT is equal to the default value, it shall be absent from the encoding.

The components of a set type shall be encoded in an order determined as follows:

a) those with universal class tags shall be encoded first, followed by those with application class tags, followed by those with context-specific tags, followed by those with private class tags;

b) within each class of tags, the components shall be encoded in the ascending order of their tags.

The components of a set-of value shall be encoded in ascending order of the octet values of their encodings.

G.6 *Boolean type*

If the value of a boolean type is true, the encoding shall have its contents octets set to hexadecimal FF.

G.7 *Real type*

The encoding of a real type which is base 2 related shall proceed as follows:

a) The number shall be represented as

S*M*2**E

Where S is plus or minus one, M is zero or is an odd integer, and E is a positive or negative integer or zero.

b) The value shall then be encoded with no change to M or E, and with F set to zero. The exponent shall be encoded in the minimum number of octets, and there shall be no leading zeros in M.

ANNEX H

(to Recommendation T.415)

(informative)

Use of the Distinguished Encoding type

H.1 *The problem to be solved*

The Distinguished Encoding has been provided to assist in the Provision of integrity security mechanisms using authenticators for material to be transferred.

The concept of an authenticator is well understood, and involves taking the bit pattern to be transferred, applying some form of hashing function to it to reduce it to a few octets, encrypting those octets to authenticate the authenticator, then transmitting the authenticator with the original material (the original material being sent in clear). On receipt, the authenticator is recalculated from the received clear text and compared with the received authenticator. If they are equal, the text was not tampered with, otherwise it was.

This simple concept becomes more difficult when the ISO model, and particularly the presentation layer, is in use.

Two problems arise, one of which is a question of modeling and so-called layer independence, and the second of which relates to the use of application layer relays, such as are used in X.400.

On the modeling issue, the hash function and the encryption algorithm are part of the application's operation, but the application has no knowledge or control of the actual encoding which the presentation layer will use. Similarly on receipt, decoding, and hence destruction of the bit-string on receipt is a presentation layer matter. There are three solutions that have been proposed to overcome this problem:

a) rule out of order the use of the actual octets produced by presentation layer for use in the authenticator; (the current philosophy being adopted by presentation and ULA experts);

b) put the hashing and authenticator mechanisms into the presentation layer itself (this solution was rejected as part of the broad question of putting support for encryption into ASN.1; at the time of rejection, the reason for the rejection was that work in security was still immature, and that one did not want to prejudice the eventual result);

c) model a complex interaction with the presentation layer where, on transmission, a value is presented for encoding, the encoding is produced and returned to the application layer which calculates the authenticator, then the whole is transmitted; on receipt, as well as producing the abstract value, the received encodings are passed to the application layer for authenticator checking; (this model was rejected by the ULA group);

d) do the entire encoding in the application layer, and make no use of presentation services for transfer syntax negotiation; (this is really a rejection of the OSI reference model, and would not be acceptable as a wide-spread solution).

It might be argued that failure to agree on a model to describe an apparently simple and workable process (produce the encoding, then the authenticator, and transmit both, check against the authenticator on receipt) is not something which should be accepted as a long-term position. Such a remark would have strong validity if it were not for the second problem of application relays, and if there were no other workable solution. (This document is outlining an alternative solution which is used in X.500 and is considered to be free from modeling and relay-system problems and workable).

The second problem is that, if an application relay is in place, the transfer syntax used for the second transmission may be different from that agreed for the first (for example, use of packed encoding rules on one of them, and Basic Encoding Rules on the other). This would defeat the authenticator, unless the authenticator was opened up and recalculated at the relay, which would imply security exchanges with the relay, whereas what is required is end-to-end security.

Note – There have been suggestions that one might want to flag a presentation context as “do not decode/re-encode at application relays”, but this also provides modeling and other problems.

Thus we are led to try to work with a model in which the presentation layer (together with any intervening application relays) provides for the transfer of the abstract syntax and semantics of the information, but makes no guarantees that the actual bit-pattern encoding (the transfer syntax) will be retained end-to-end.

The challenge is therefore to provide an authenticator mechanism that can operate on the abstract datatype, rather than on the transmitted bit string.

The Directories group were the first to attempt to produce a solution to this problem, and it is their model that is described below.

H.2 *The approach to a solution*

The following text describes first a conceptual model of what is being done, followed by an implementation optimization that eliminates the double encoding/decoding implied in the conceptual model.

The conceptual model works as follows:

a) The sender, in the application layer, converts the abstract syntax value into a bitstring using the Distinguished Encoding Rules, and produces the authenticator from that bitstring, which is inserted added to the abstract syntax value, and both values transmitted using normal presentation layer mechanisms, and any transfer syntax. Conceptually, the sender is encoding twice – once for the authenticator (using the Distinguished Encoding Rules) in the application layer, and once for the actual transfer (using the negotiated transfer syntax) in the presentation layer.

   Note – The import property of the bitstring produced by the Distinguished Encoding Rules is that it is in one-to-one correspondence with the abstract value. Thus end-to-end transfer without loss of information at the abstract syntax value is equivalent to end-to-end transfer of the bitstring on which the authenticator is based.

b) The receiver will decode the received bitstring in the presentation layer, using the negotiated transfer syntax (which may differ from that used by the sender of an application relay is in place), and will pass the abstract value to the application. In the application layer, the abstract value is re-encoded using the Distinguished Encoding Rules to produce the bitstring to be authenticated.

Thus conceptually, we encode twice at the sending end, and decode once and then encode at the receiving end. Implementors may choose to actually do this if the code supporting presentation layer operation is from a supplier different from that producing the code to support the application. How significant an overhead this would be is not clear at this stage. Where an integrated implementation is used, however, there is the option of the optimization described below. It should also be noted that the Distinguished Encoding Rules are no harder to apply than the Basic Encoding Rules except in relation to use of set-of. If a large set-of is to be processed, the implementation may need to invoke a disk-based sorting routine. Application designers should be aware of this, and try to use sequence-of instead of set-of when use of the Distinguished Encoding Rules is envisaged.

H.3 *The implementation optimization*

The OSI model and protocol standards specify required behaviour, they do not, in any way, seek to constrain the architecture and structure of actual implementation code. Thus an implementor can produce the desired effect in whatever way he chooses.

At the sending end, the bitstring which is produced (conceptually in the application layer) can be kept, and used to provide the encoding that is conceptually performed in the presentation layer. This is suitable for sending if the negotiated transfer syntax is either the ASN.1 Basic Encoding Rules or the ASN.1 Distinguished Encoding Rules. If it is neither of these, then double encoding is necessary.

Similarly at the receiving end, the received bitstring can be retained (for any transfer syntax), and the implementation can use this to check the authenticator. If it matches, end of problem, if it does not match, then it may be a transfer syntax problem, and recoding from the abstract value is necessary to determine whether there was tampering or not.

In order to maximize the chances of not having to do double encoding/decoding, systems using this mechanism would be advised to try to negotiate a transfer syntax of Distinguished Encoding Rules (using the appropriate ASN.1 Object Identifier) as their first preference, falling back onto Basic Encoding Rules (first preference) or Packed or some other encoding rules second preference.