### The ALGORITHM information object class

The information object class concept is also used for defining cryptographic algorithms. The ALGORITHM information object class is defined in 6.2.2 of Rec, ITU-T X.509 | ISO/IEC 9594-8. The specification of this information is copied below for easy reference. The ALGORITHM information object class is different from most other information object classes in the sense that an instance of an information object is an invocation of the algorithm rather than specifying a value identifying something, like an e-mail address.

The following ASN.1 information object class is used for specifying cryptographic algorithms.

ALGORITHM ::= CLASS {

&Type OPTIONAL,

&DynParms OPTIONAL,

&id OBJECT IDENTIFIER UNIQUE }

WITH SYNTAX {

[PARMS &Type]

[DYN-PARMS &DynParms ]

IDENTIFIED BY &id }

The ALGORITHM information object class has the following fields:

a) The &Type field is used for specifying those fixed parameters that are necessary for specifying the exact procedure for deploying the cryptographic algorithm being defined. Not all cryptographic algorithms require such parameters. The field is then absent or has the value NULL, as determined by the individual cryptographic algorithm specifications.

b) The &DynParams field is used for specifying those dynamic parameters that determine the value(s) to be exchanged between two communicating entities when invoking the cryptographic algorithm. Not all cryptographic algorithms require dynamic parameters. In this case the &DynParams field shall be absent.

c) The &id field is used for uniquely identify the class of cryptographic algorithm being defined.

The AlgorithmWithInvoke parameterized data type defined below is used in situations where the type of cryptographic algorithm is signalled together with its invocation.

AlgorithmWithInvoke{ALGORITHM:SupportedAlgorithms} ::= SEQUENCE {

algorithm ALGORITHM.&id({SupportedAlgorithms}),

parameters [0] ALGORITHM.&Type({SupportedAlgorithms}{@algorithm}) OPTIONAL,

dynamParms [1] ALGORITHM.&DynParms({SupportedAlgorithms}{@algorithm}) OPTIONAL,

... }

The AlgorithmWithInvoke parameterized data type has the following components;

a) The algorithm component shall hold an object identifier that uniquely identify the cryptographic algorithm being defined.

b) The parameters component, when present, shall hold the values of the fixed parameters that further identify the cryptographic algorithm in question. This component shall be present when the &Type field is present in the information object for the cryptographic algorithm in question. Otherwise, it shall be absent.

c) The dynamParms component, when present, shall hold the value(s) required by the dynamic parameters for the cryptographic algorithm. This component shall be present when the &DynParams field is present in the information object for the cryptographic algorithm. Otherwise, it shall be absent.

The AlgorithmIdentifier parameterized data type defined below is used in situations where the type of cryptographic algorithm is signalled without a corresponding invocation.

AlgorithmIdentifier{ALGORITHM:SupportedAlgorithms} ::= SEQUENCE {

algorithm ALGORITHM.&id({SupportedAlgorithms}),

parameters ALGORITHM.&Type({SupportedAlgorithms}{@algorithm}) OPTIONAL,

... }

The components of AlgorithmIdentifier data type shall be as specified as for the corresponding components of the AlgorithmWithIvoke parameterized data type.

The AlgoInvoke parameterized data type defined below is used when the cryptographic algorithm has previously been determined and where only invocation information is required.

AlgoInvoke{ALGORITHM:SupportedAlgorithms} ::=

ALGORITHM.&DynParms({SupportedAlgorithms})