3.2 The EncryptionMethod Element

EncryptionMethod is an optional element that describes the encryption algorithm applied to the cipher data. If the element is absent, the encryption algorithm must be known ~~by~~ to the recipient or the decryption will fail.

Schema Definition:

```
<complexType name='EncryptionMethodType'
mixed='true'>

    <sequence>

      <element name='KeySize' minOccurs='0'
type='xenc:KeySizeType'/>

      <element name='OAEPparams' minOccurs='0'
type='base64Binary'/>

      <any namespace='##other' minOccurs='0'
maxOccurs='unbounded'/>

    </sequence>

    <attribute name='Algorithm' type='anyURI'
use='required'/>
    <attribute name='MGF' type='anyURI'
use='optional'/>

  </complexType>
```

The permitted child elements of the EncryptionMethod are determined by the specific value of the Algorithm attribute URI, and the KeySize child element is always permitted. For example, the RSA-OAEP algorithm (section 5.5.2 RSA-OAEP) uses the ds:DigestMethod and OAEPparams elements. (We rely upon the ANY schema construct because it is not

possible to specify element content based on the value of an attribute.)

The presence of any child element under `EncryptionMethod` that is not permitted by the algorithm or the presence of a `KeySize` child inconsistent with the algorithm must be treated as an error. (All algorithm URIs specified in this document imply a key size but this is not true in general. Most popular stream cipher algorithms take variable size keys.)

The MGF attribute is optional and may be used for specifying the Mask Generation Function for RSA-OAEP. It is defined in the xenc11: namespace.

…

5.1 Algorithm Identifiers and Implementation Requirements

…

Key Transport

1. required RSA-v1.5
   http://www.w3.org/2001/04/xmlenc#rsa-1_5
2. required RSA-OAEP (including MGF1 with SHA1)
   http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p
3. optional RSA-OAEP
   http://www.w3.org/2001/04/xmlenc#rsa-oaep

…

5.5 Key Transport

Key Transport algorithms are public key encryption algorithms especially specified for encrypting and decrypting keys. Their identifiers appear as `Algorithm` attributes to `EncryptionMethod` elements that are children of `EncryptedKey`. `EncryptedKey` is in turn the child of a `ds:KeyInfo` element. The type of key being transported, that is to say the algorithm in which it is planned to use the transported key, is given by the `Algorithm` attribute of the `EncryptionMethod` child of the `EncryptedData` or `EncryptedKey` parent of this `ds:KeyInfo` element.

(Key Transport algorithms may optionally be used to encrypt data in which case they appear directly as the `Algorithm` attribute of an `EncryptionMethod` child of an `EncryptedData` element. Because they use public key algorithms directly, Key Transport algorithms are not efficient for the transport of any amounts of data significantly larger than symmetric keys.)

~~The RSA v1.5 Key Transport algorithm given below are those used in conjunction with TRIPLEDES and the Cryptographic Message Syntax (CMS) of S/MIME [*CMS-Algorithms*]. The RSA v2 Key Transport algorithm given below is that used in conjunction with AES and CMS [*AES-WRAP*].~~

5.5.1 RSA Version 1.5

Identifier:
http://www.w3.org/2001/04/xmlenc#rsa-1_5 (required)

The RSAES-PKCS1-v1_5 algorithm, specified in RFC 3447 [*PKCS1*], takes no explicit parameters. An example of an

RSA Version 1.5 `EncryptionMethod` element is:

```
<EncryptionMethod

 Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-
1_5"/>
```

The `CipherValue` for such an encrypted key is the base64 [*RFC2045*] encoding of the octet string computed as per RFC 3447 [*PKCS1*], section 7.2.1: Encryption operation]. As specified in the EME-PKCS1-v1_5 function RFC 3447 [*PKCS1*], section 7.2.1, the value input to the key transport function is as follows:

```
CRYPT ( PAD ( KEY ))
```

where the padding is of the following special form:

```
02 | PS* | 00 | key
```

where "|" is concatenation, "02" and "00" are fixed octets of the corresponding hexadecimal value, PS is a string of strong pseudo-random octets [*RANDOM*] at least eight octets long, containing no zero octets, and long enough that the value of the quantity being CRYPTed is one octet shorter than the RSA modulus, and "key" is the key being transported. The key is 192 bits for TRIPLEDES and 128, 192, or 256 bits for AES.

Implementations must support this key transport algorithm for transporting 192-bit TRIPLEDES keys. Support of this algorithm for transporting other keys is optional. RSA-OAEP is recommended for the transport of AES keys.

The resulting base64 [*RFC2045*] string is the value of the child text node of the `CipherData` element, e.g.

```
<CipherData>
```

```
    <CipherValue>IWijxQjUrcXBYoCei4QxjWo9Kg8D3p9tlWoT4
     t0/gyTE96639In0FZFY2/rvP+/bMJ01EArmKZsR5VW3rwoPxw=
    </CipherValue>
  </CipherData>
```

5.5.2 RSA-OAEP

Identifier:
http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p
~~(required)~~
Identifier:http://www.w3.org/2001/04/xmlenc#rsa-oaep

The RSAES-OAEP-ENCRYPT algorithm, as specified in RFC 3447 [*PKCS1*], ~~takes~~ has options that define the message digest function and mask generation function, as well as an optional PSourceAlgorithm parameter~~three parameters~~. ~~The~~ Default values defined in RFC 3447 are SHA1 for the message digest and MGF1 with SHA1 for the mask generation function. ~~two user specified parameters are a MANDATORY message digest function and an optional encoding octet string `OAEPparams`.~~ Both the message digest and mask generation functions are used in the EME-OAEP-ENCODE operation as part of RSAES-OAEP-ENCRYPT.

The http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p identifier defines the mask generation function as the fixed value of MGF1 with SHA1. In this case the optional xenc11:MGF attribute of the xenc:EncryptionMethod element MUST NOT be provided.

The http://www.w3.org/2001/04/xmlenc#rsa-oaep identifier

defines the mask generation function using the value of the optional xenc11:MGF attribute of the xenc:EncryptionMethod element. If not present, the default of MGF1 with SHA1 is to be used.

Otherwise the two identifiers define the same usage of the RSA-OAEP algorithm, as follows.

1. ~~specifiesare~~The~~The~~ message digest function ~~is~~ SHOULD be specified using ~~indicated by~~ the Algorithm attribute of the~~a child~~ ds:DigestMethod child element of the xenc:EncryptionMethod element. If it is not specified, the default value of SHA1 is to be used.~~provided~~

The optional RSA-OAEP PSourceAlgorithm parameter value MAY be explicitly provided by placing the base64 encoded octets in the xenc:OAEPparams XML element.~~and the mask generation function, the third parameter, is always MGF1 with SHA1 (mgf1SHA1Identifier). Both the message digest and mask generation functions are used in the EME-OAEP-ENCODE operation as part of RSAES-OAEP-ENCRYPT.~~

~~The encoding octet string is the base64 decoding of the content of an optional `OAEPparams` child element . If no `OAEPparams` child is provided, a null string is used.~~

Schema Definition:

```
     <!-- use these element types as children of
EncryptionMethod

           when used with RSA-OAEP -->
```

```
      <element name='OAEPparams' minOccurs='0'
type='base64Binary'/>

      <element ref='ds:DigestMethod' minOccurs='0'/>
```

An example of an RSA-OAEP element is:

```
  <EncryptionMethod

      Algorithm=http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p
      MGF="
http://www.w3.org/2001/04/xmlenc#MGF1withSHA1">

     <OAEPparams>9lWu3Q==</OAEPparams>

     <ds:DigestMethod


Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

  <EncryptionMethod>
```

The `CipherValue` for an RSA-OAEP encrypted key is the base64 [*RFC2045*] encoding of the octet string computed as per RFC 3447 [*PKCS1*], section 7.1.1: Encryption operation. As described in the EME-OAEP-ENCODE function RFC 3447 [*PKCS1*], section 7.1.1, the value input to the key transport function is calculated using the message digest function and string specified in the `DigestMethod` and `OAEPparams` elements and using the mask generator function ~~MGF1 (with SHA1) specified in RFC 3447~~as specified with the MGF attribute or the default. The desired output length for EME-OAEP-ENCODE is one byte shorter than the RSA modulus.

The transported key size is 192 bits for TRIPLEDES and 128, 192, or 256 bits for AES. Implementations must implement RSA-OAEP for the transport of all key types and sizes that are mandatory to implement for symmetric encryption~~128 and 256 bit keys~~. They may implement RSA-OAEP for the transport of other keys.