Encrypted DigiDoc Format Specification

Document Ver: 1.1

Format Specification Ver: 1.0

Modified: 25.06.2012

Document Versions

Document Specifications	
Created	14.03.2005
Title	Encrypted DigiDoc Format Specification
Contracting Authority	AS Sertifitseerimiskeskus (Certification Centre Ltd.)
Author	Veiko Sinivee, Kristi Uukkivi
Version	1.1

Version Specifications		
Date	**Version**	**Modifications**
14.03.2005	1.0	Document created
25.06.2012	1.1	Added chapter regarding the use of padding methods

Contents

This document describes the document format, that is used by DigiDoc applications for the encryption of data (hereinafter: ENCDOC-XML). Encrypted DigiDoc documents are presented in XML according to the international XML-ENC standard.

1. References

XML-ENC	XML Encryption Syntax and Processing. W3c Recommendation 10 December 2002. http://www.w3.org/TR/xmlenc-core/.
RFC3275	Eastlake 3rd D., Reagle J., Solo D., (Extensible Markup Language) XML-Signature Syntax and Processing. (XML-DSIG) March 2002.
XML Schema 2	XML Schema Part 2: Datatypes. W3C Recommendation 02 May 2001 http://www.w3.org/TR/xmlschema-2/
DAS	Digital Signature Act of the Republic of Estonia
X.923	Padding method complying with ANSI X.923 standard http://en.wikipedia.org/wiki/Padding_%28cryptography%29#Byte_padding
PKCS#7	Cryptographic Message Syntax Standard. http://www.rsa.com/rsalabs/node.asp?id=2129
DigiDoc	http://www.id.ee/index.php?id=30287

2. Introduction

XML Encryption Syntax and Processing (XML-ENC) defines the formatting, that enables structural presentation of encrypted data, and related security attributes (eg. transport keys).

This document describes the formatting of DigiDoc encrypted documents, according to the XML-ENC standard, and is a sub-set of that standard. Digidoc encrypted document format (ENCDOC-XML) is a type of profile of XML-ENC that is in strict compliance with the XML-ENC standard, but the DigiDoc Library does not support all types of formats specified in the XML-ENC standard (eg. DES, 3DES, encryption of the contents of a XML element etc.).

Encrypted document format ENCDOC-XML was created for the encryption of integrated data sets in order to enable secure transfer to the recipient or recipients, who can decrypt the received message using their Estonian ID card or equivalent encryption devices. Thus ENCDOC-XML does not enable the encryption of the content of just any element of a XML document (without start and end tags - XML-ENC „content“ formatting) or the encryption of just a part of a XML document (XML-ENC „element“ formatting). ENCDOC-XML supports only the encryption of the whole XML document or any other binay file. The only supported encryption algorithm is the AES (Advanced Encryption Standard or Rjindael) 128 bit transport keys, that are encrypted pursuant to the RSA 1.5 standard using the recipient´s certificate. The XML-ENC standard provides several possibilities for the the assignment of the recipient of the encrypted transport key:

- EncryptedKey – Recipient attribute
- EncryptedKey – KeyName sub-element
- EncryptedKey – CarriedKeyName sub-element
- adding the recipient´s certificate

Since all mentioned elements are non-compulsory and depend on the application used, then we only use the last option, i.e. the recipient´s certificate must be added for each recipient separately. The certificate contains information about afore mentioned data and its format, and its meaning is contained in the certificate, and therefore it does not depend on specific applications. These non-compulsory elements are also allowed in the ENCDOC-XML format, and libraries, as well as applications, should support them, but not as a requirement nor depend on their existence.

The DigiDoc encrypted document format has the following critical features:

- enables the encryption of only one data set. If you need to store several different data sets in one encrypted file, then those data sets should be stored in a XML file in DIGIDOC-XML format, and then encrypted;
- enables the creation of encrypted transport keys for one or more recipients;
- the data set chosen for encryption may be a XML or any other binary file (Word, Excel, PDF, RTF etc.)
- enables the identification of the version of the library used to create the document and distinguish between possible new document formats that might be created in the future;
- this document format does not add any new XML elements to the existing XML-ENC format, because it utilizes other options provided by the standard;
- the ZLIB algorithm enables the compression of original data prior to encryption.

The obligatory elements and attributes provided in the XML-ENC specification have been integrated without any modifications. ENCDOC-XML formatting contains those non-obligatory XML-ENC elements, that enable the identification of the suitable key for the decryption of the recipient or display recipient´s data in the application. cdoc extension is used in order to distinguish files encrypted in DigiDoc format.

3. General Structure of ENCDOC-XML Documents

The structure of ENCDOC-XML files is the following (notation defined in [RFC3275] chapter 2):

```
<?xml version="1.0" encoding="UTF-8" ?>
<denc:EncryptedData Id=? Type=? MimeType=? xmlns:denc= >
  <denc:EncryptionMethod Algorithm= />
  <ds:KeyInfo xmlns:ds= >
<!-- one encrypted transport key for each recipient -->
    <denc:EncryptedKey Id=? Recipient=? xmlns:denc= >
      <denc:EncryptionMethod Algorithm= />
      <ds:KeyInfo xmlns= >
        (<ds:KeyName/>)?
  <!-- recipient´s certificate -->
        <ds:X509Data>
          <ds:X509Certificate />
        </ds:X509Data>
        (<ds:CarriedKeyName/>)?
      </ds:KeyInfo>
      <denc:CipherData>
  <!-- RSA 1.5 encrypted transport key -->
        <denc:CipherValue/>
      </denc:CipherData>
    </denc:EncryptedKey>
  </ds:KeyInfo>
<!-- encrypted Data -->
    <denc:CipherData>
        <denc:CipherValue/>
    </denc:CipherData>
<!-- meta-information about the document and creator -->
  <denc:EncryptionProperties Id=?>
    (<denc:EncryptionProperty Id=? Target=? Name= />)+
  </denc:EncryptionProperties>
</denc:EncryptedData>
```

Thus the ENCDOC-XML is a container <EncryptedData /> containing encrypted data and transport keys.

.

Encrypted Data – encrypted and compressed data in the form of Base64.

Transport keys – original data is encrypted using only one 128 bit AES key, which is encrypted with an individual certificate corresponding to the possible recipient/decryptor of each document. The structure of the transport key or <EncryptedKey> contains the AES transport key encrypted with the relevant person´s certificate, and the relevant certificate itself, which is used to identify the relevant private key needed for the decryption of the transport key.

4. Elements and Parameters

4.1. Root Element (EncryptedData)

Each encrypted DigiDoc file has a root element **<EncryptedData>** with the following attributes:

ID – unique marker of the element. ID attributes can be used in encrypted documents, but it is not required and applications should not presume their existence.

Type – a non-compulsory attribute, that indicates whether encrypted data constitute the content of some XML element (Type="http://www.w3.org/2001/04/xmlenc#Content“) or whether it contains start end end tags of the element (Type="http://www.w3.org/2001/04/xmlenc#Element“). This attribute is not used here, because encrypted DigiDoc files may contain binary data.

MimeType – refers to the type of encrypted data. If the data was compressed using the ZLIB algorithm prior to encryption, then in encrypted DigiDoc files this attribute designated the value „http://www.isi.edu/in- noes/iana/assignments/media-types/application/zip“. In addition, mime type „http://www.sk.ee/DigiDoc/v1.3.0/digidoc.xsd“ is used in case the original data is a DigiDoc (digitally signed) document. If the data is compressed prior to encryption, then the initial mime type (if it was used) is saved in the sub-element <EncryptionProperty>, which will be subsequently discussed.

xmlns - must use XML-ENC namespace: http://www.w3.org/2001/04/xmlenc#. Encrypted DigiDoc files use XML-ENC namespace abbreviation „denc“, which is added to all elements from the relevant namespace, i.e. <denc:EncryptedData>. In encrypted XML documents each element <EncryptedData> always has a sub-element.

EncryptionMethod – the attribute´s **algorithm** value indicates the algorithm used for the encryption of data. For the time being only 128 bit AES transport keys are used in encrypted DigiDoc files, and therefore the value of the attribute is always "http://www.w3.org/2001/04/xmlenc#aes128-cbc".

4.2. Transport Keys (EncryptedKey)

Each encrypted DigiDoc file contains one or more transport keys. The root element <EncryptedData> contains sub-element <KeyInfo>, which in turn contains all elements <EncryptedKey>. Each such element contains at least:

- element <EncryptionMethod>, with attribute algorithm value „http://www.w3.org/2001/04/xmlenc#rsa-1_5“ and indicates that the value of the AES transport key is encrypted with the corresponding recipient´s public certificate using the RSA 1.5 method.
- corresponding recipient´s certificate in the form of –base64 in sub-element <ds:KeyInfo><ds:X509Data><ds:X509Certificate>.
- the real encrypted AES transport key in the form of –base64 in sub-element <denc:CipherData><denc:CipherValue>.

In addition, the element <EncryptedData> may feature:

- ID attribute – unique marker
- recipient attribute – some readable name or marker for the recipient. Applications often use the CN attribute on the DN field of the recipient´s certificate.
- Key Name – in sub-element <ds:KeyInfo><ds:KeyName>
- Other Key Name - in sub-element <ds:KeyInfo><ds:CarriedKeyName>

Above mentioned additional attributes can be used for connecting information displayed on screen with the relevant transport key, but since these are non-compulsory elements, then applications should not depend on their use or stipulate requirements regarding their content. XML-ENC standard allows for references and meta information (<EncryptionProperty>) in the transport key, which is not supported by the encrypted DigiDoc format. XML-ENC standard enables the moving of transport keys within the document, and connection of the keys with encrypted data via special reference elements or ID attributes, that the encrypted DigiDoc format also does not support. This format presumes that all transport keys are contained in the <EncryptedData> sub-element <KeyInfo>.

4.3. *Encrypted Data*

The data in encrypted using the 128 bit AES transport keys and saved in the form of base64 <EncryptedData> in sub-element <CipherData>/<CipherValue>.

4.4. *Meta Data*

Encrypted DigiDoc documents may contain meta data in element <EncryptionProperties>.

This element contains one or more <EncryptionProperty> elements, that contain saved information about the initial data of some information.

Element <EncryptionProperties> may itself have the attribute:

- ID – unique marker

Each sub-element <EncryptionProperty> may have:

- ID attribute – unique marker
- Target attribute – reference to the element´s <EncryptedData> ID attribute in the form of „#<id-value>“
- Any other attributes and real content. In the encrypted DigiDoc format a special attribute **Name** is defined, and the value of this name is used to distinguish between special types of meta data.
  - Name="LibraryVersion" - contains the name and version of the library used to create the relevant document in the form of „<name>|<version>“.
  - Name="DocumentFormat" - contains the name and version of the encrypted

    DigiDoc document format in the form of „<name>|<version>“. The current relevant format and version is „ENCDOC-XML|1.0“.
  - Name="Filename" - contains the original file name of the encrypted document.
  - Name="OriginalSize" - contains the initial byte size of the encrypted document.
  - Name="OriginalMimeType" - contains the original mime type of the encrypted document. It is used only if the data was compressed using the ZLIB algorithm prior to encryption, and thus the previous mime type had to be removed from the attribute <EncryptedData MimeType=""> and the attribute was used before. For example, in case of encrypted and compressed DigiDoc documents the element always contains the mime type of the digitally signed DigiDoc document: „http://www.sk.ee/DigiDoc/v1.3.0/digidoc.xsd“.
  - Name="orig_file" - is used only in case the encrypted data is a digitally signed DigiDoc file. In that case each DigiDoc document datafile (element <DataFile>) is furnished with the following entry: „<file-name>|<size- in-bytes>|<mime-type>|<datefile-id-attribute>“.

5. The Padding Method

In CDigiDoc and JDigiDoc libraries PKCS#7 and X.923 standard padding methods are used for the creation of encrypted DigDoc files.

The padding method was modified as of CDigiDoc library version 3.5.1 and JDigiDoc library version 3.6.1.1 (both released in April 2012).

Using the padding method with CDigiDoc library versions before 3.5.1 and JDigiDoc library versions before 3.6.1.1:

- PKCS#7 padding is always added by systemic libraries (in case of CdigiDoc it´s the OpenSSL library; in case of JDigiDoc it´s the Bouncy Castle library);
- X.923 padding is added by CDigiDoc and JDigiDoc libraries on certain conditions, but not always. Libraries differ in the default use of padding.

Using the padding method with CDigiDoc 3.5.1 and JDigiDoc 3.6.1.1 and subsequent library versions:

- PKCS#7 padding is always added by systemic libraries;
- X.923 padding is always added by CDigiDoc and JDigiDoc libraries.

Note: Encrypted files created with DigiDoc libraries using the padding method modified as of April 2012 may in special circumstances not be compatible with prior versions of these libraries.