BDOC – FORMAT FOR DIGITAL SIGNATURES

Version 2.1.2:2014

OID: 1.3.6.1.4.1.10015.1000.3.2.3

Table of Contents

Introduction

The European Directive on a community framework for Electronic Signatures defines an electronic signature as: "data in electronic form which is attached to or logically associated with other electronic data and which serves as a method of authentication".

The present document is intended to cover electronic signatures for various types of transactions, including business transactions (e.g. purchase requisition, contract, and invoice applications). Thus the present document can be used for any transaction between an individual and a company, between two companies, between an individual and a governmental body, etc. The present document is independent of any environment. It can be used with different signature creation devices e.g. smart cards, GSM SIM cards, special programs for electronic signatures, etc.

The ETSI standard TS 101 903 [1] (hereinafter: XAdES) defines formats for advanced electronic signatures that remain valid over long periods and incorporate additional useful information in common use cases (like indication of the role or resolution of the signatory). XAdES is XML-based and therefore suitable for the current ICT environment. The ETSI standard TS 103 171[8] further profiles the XAdES signature by putting limitations on choices.

The ETSI standard TS 102 918[9] (hereinafter: ASiC) defines format of container for encapsulation of signed files and signatures with extra information. The ETSI TS 103 174[10] profiles in further on.

This BDOC specification is fully compliant with abovementioned ETSI standards.

The present document:

- specifies profiles of XAdES by narrowing down choices of elements and value types in the standard;
- defines sets of XAdES elements for long-time validity of XAdES signature;
- specifies container format for embedding signed files and signatures based on ASiC.

For the further reference, term BDOC is used thorough the text to denote both XAdES profile and container format.

1. Scope

The present document defines XML formats for advanced electronic signatures that remain valid over long periods and incorporate additional useful information in common uses cases. This includes evidence as to its validity even if the signer or verifying party later attempts to deny (repudiates) the validity of the signature.

The present document builds on the following standards:

- ETSI TS 101 903 v1.4.2 – XML Advanced Electronic Signatures (XAdES) [1] and its Baseline Profile ETSI TS 103 171[8];
- ITU-T Recommendation X.509 [2];
- RFC 3161 – PKIX Time-Stamp protocol [3];
- RFC 6960 – Online Certificate Status Protocol [4];
- ETSI TS 102 918 v1.2.1 - Associated Signature Containers (ASiC) [9] and its Baseline Profile ETSI TS 103 174[10]. The latter is in turn based on OpenDocument [5] standard part *OpenDocument-v1.2-part3 – Packages*.

For a complete list of references, see section 2.

Section 5 defines basic profile of the BDOC format. This profile contains just signature without any validation data.

Section 6 defines two profiles of the BDOC format with validation data providing for "replacement of handwritten signature".

Section 7 discusses and defines means for achieving long-time validity of the electronic signatures.

Section 8 specifies container format for embedding signed files and signatures into one data unit.

2. References

[1] ETSI TS 101 903 V1.4.2 (2012-10) – XML Advanced Electronic Signatures (XAdES)

[2] ITU-T Recommendation X.509: "Information technology - Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks"

[3] IETF RFC 3161: "Internet X.509 Public Key Infrastructure Time-Stamp protocol"

[4] RFC 6960: “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP”

[5] OASIS "Open Document Format for Office Applications (OpenDocument) Version 1.2 Part 3: Packages"

[6] IETF RFC 3275: “XML-Signature Syntax and Processing”

[7] ETSI TS 102 023 V1.2.2 (2008-10) – Policy requirements for time-stamping authorities

[8] ETSI TS 103 171 V2.1.1 (2012-03) - XAdES Baseline Profile

[9] ETSI TS 102 918 V1.2.1 (2012-02) - Associated Signature Containers (ASiC)

[10] ETSI TS 103 174 V2.1.1 (2012-03) - ASiC Baseline Profile

[11] ETSI TS 102 176-1 V2.1.1 (2011-07) – Algorithms and Parameters for Secure Electronic Signatures; Part 1: Hash functions and asymmetric algorithms

[12] RFC 5280: “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”

3. Definitions and Abbreviations

See Clause 4 of XAdES[1] for basic definitions and abbreviations.

BDOC – a profile of XAdES and container packaging rules

4. Overview

Whereas XAdES has been around for several years and there are a number of implementations of this standard around, they remain incompatible. The reasons are the following:

- XAdES allows for myriad of options. Implementations of XAdES usually do not support every non-mandatory building block or element which results in incompatibility of XAdES signatures;
- Use of XAdES optional building blocks heavily depends on security requirements of application and PKI services provided. As those requirements and set of services tend to vary, corresponding XAdES profiles do as well.
- XAdES specifies just a signature format allowing the source data (to be signed) be anywhere and referenced by URI. In practice it is often required source data and signatures to be bound together in a single data unit ("container" or "file"). Implementers have free choice here which results in incompatibility of *digitally signed files*.

Recent additions to ETSI standards have addressed the problem by introducing XAdES Baseline Profile[8] and by standardizing container format[9] and Baseline Profile of thereof[10].

This specification uses new base standards and solves abovementioned problems by:

- defining subset of XAdES elements and parameters – "BDOC profile of XAdES";
- defining requirement profiles for PKI, time-stamping and certificate validation services and corresponding XAdES building blocks;
- defining container format for embedding source data and signatures – "BDOC file format".

This document is based entirely on XAdES[1] and therefore is not self-consistent. The reader shall use this standard as a basis and follow references and profiling notes given in this document. Requirements from XAdES Baseline Profile[8], ASiC[9] and its Baseline Profile[10] shall be covered by this specification but could give reader expanded insight..

Appenix 2 contains overview of use of XAdES element in different BDOC variations.

5. BDOC Basic Profile

The BDOC Basic Profile is an XML structure containing a single cryptographic signature over the well-defined set of data. It does not contain any validation data for full signature validation such as timestamps or certificate validity confirmations. It just forms basis for other forms of BDOC described in next chapter.

The BDOC Basic profile bases on Basic Electronic Signature – XAdES-BES – defined in clause 4.4.1 of XAdES [1].

Following notions will be used in further text for defining requirements for usage of the elements:

Notion	***Creating application***	***Processing application***
M (Mandatory)	Must produce this element	Must process this element
C (Critical)	May produce this element	Must process this element if present
O (Optional)	May produce this element	May process this element if present
N/A	Element is not used	Element is not used

Next, we profile `xades:Cert` element for use in both in BDOC Basic Profile and BDOC extended forms described in next chapter.

5.1. Use of Cryptographic Algorithms

Contemporary sources shall be consulted when selecting cryptographic algorithms and their key lengths. Good sources are relevant ETSI standard[11] and www.keylength.com. The following are recommendations valid at the time of creation of this document. BDOC specification does not restrict use of other cryptographic algorithms and key lengths.

Hash Algorithms
SHA-256 or better is strongly recommended when creating BDOC documents. However, due to technical difficulties it is not always possible to use anything else besides SHA-1. As a consequence, the BDOC-compatible application must be capable of handling SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512 algorithms in verification and make best efforts to create BDOC with SHA-256 or better. Supported values of the URI in `DigestMethod` elements for parameter `Algorithm` are thus:

```
http://www.w3.org/2000/09/xmldsig#sha1
http://www.w3.org/2001/04/xmldsig-more#sha224
http://www.w3.org/2001/04/xmlenc#sha256
http://www.w3.org/2001/04/xmldsig-more#sha384
http://www.w3.org/2001/04/xmlenc#sha512
```

Asymmetric Key Algorithms
The algorithm and key length are solely determined by abilities of cryptographic device when creating a BDOC document. The recommendation is to use at least 2048-bit key in case of RSA and at least 224-bit key length when in case of using elliptic curves (ECDSA). Allowed values for the URI in the element `SignatureMethod` are thus:

```
http://www.w3.org/2000/09/xmldsig#rsa-sha1
http://www.w3.org/2001/04/xmldsig-more#rsa-sha224
http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
http://www.w3.org/2001/04/xmldsig-more#rsa-sha384
http://www.w3.org/2001/04/xmldsig-more#rsa-sha512


http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha1
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha224
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha384
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha512
```

5.2. Definition of BDOC Base Profile

The BDOC Basic Profile structure consists of:

- `ds:SignedInfo` block containing references (with hash values) to data objects to be signed
- `ds:SignatureValue` element containing the actual signature value
- `ds:KeyInfo` structure containing signer's certificate
- Additional data in `xades:QualifyingProperties` block up to XAdES-EPES level.

Following restrictions apply for elements of `ds:SignedInfo` block:

Element		***Parameter***	***Comment***
Signature	M	Id	For example "S0"
SignedInfo	M		
CanonicalizationMethod	M	Algorithm="http://www.w3.org/2006/12/xml-c14n11"	
SignatureMethod	M	Algorithm=" http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"	See section 5.1
Reference	M	Id	See below

The `ds:SignedInfo` block contains two or more `ds:Reference` structures pointing to data objects to be signed:

- one reference to every original file in the container to be signed. In this case element `ds:Reference` must have parameter `URI` referring to the data object (e.g. "`doc.txt`").

- just one reference to `SignedProperties` data block described below. In that case element `ds:Reference` must have parameter `Type=`http://uri.etsi.org/01903/#SignedProperties and parameter `URI` referring to the SignedProperties block (e.g. "`#S0-SignedProperties`")

Element `Transforms` in `ds:Reference` structure is not created but must be treated as *Critical* if present. Element `ds:DigestMethod` shall have value
`http://www.w3.org/2001/04/xmlenc#sha256`
Section 5.1 gives further details on hash algorithms.

This specification does not mandate a separate `ds:Reference` element for `ds:KeyInfo` as the element `SigningCertificate` is mandatory by this specification and thus the signing certificate is secured by the signature.

Signers certificate shall be present in `ds:KeyInfo` element. Subelements `ds:X509Data` and `ds:X509Certificate` shall be present according to clause 4.4 of XMLDSIG [6].

XAdES[1] defines a mechanism for encapsulating additional parameters into signature by using `ds:Object` method. The `SignedProperties` element referenced in the `ds:SignedInfo` block is encapsulated in the `ds:Object` element as described in clause 6.2.1 of XAdES. Direct encapsulation is used according to XAdES clause 6.3, i.e. element `QualifyingPropertiesReference` is not supported

The following profiles use of the qualified properties of the BDOC Basic Profile:

Element		***XAdES clause***	***Comment***
QualifyingProperties	M	6.2	Must have parameter `Target` pointing to `ds:Signature` element (e.g. "`#S0`")
SignedProperties	M	6.2.1	Must have parameter `Id` (e.g. "`S0-SignedProperties`")
SignedSignatureProperties	M	6.2.3	
SigningTime	M	7.2.1	Zulu time zone shall be used
SigningCertificate	M	7.2.2	Contains only certificate of the signer; parameter `URI` is not used
SignaturePolicyIdentifier	M or N/A	7.2.3	Refers to this document. Used only in case of BDOC with time-marks (see section 6.1).
SignaturePolicyId	M	7.2.3	
SigPolicyId	M	7.1.2	Method OIDAsURN is used, value of the OID is 1.3.6.1.4.1.10015.1000.3.2.3

SigPolicyHash	M	7.2.3	Hash value of this specification; it is not mandated to verify it when validating signature
SigPolicyQualifiers	M	7.2.3	
SigPolicyQualifier SPURI	M	7.2.3.1	URL to this document
SignatureProductionPlace	C	7.2.7	
SignerRole	C	7.2.8	The `ClaimedRoles` element is allowed, `CertifiedRoles` is not supported.
SignedDataObjectProperties	M	6.2.4	
DataObjectFormat	M	7.2.5	Mandatory for every signed data object excluding element `SignedProperties`
MimeType	M	7.2.5	MIME type
ObjectReference	M	7.2.5	Reference to Id value used in `Reference` element.
CommitmentTypeIndication	N/A	7.2.6	Use `ClaimedRoles` element for indication of commitment or resolution in free text.
AllDataObjectsTimeStamp	N/A	7.2.9	
IndividualDataObjectsTimeStamp	N/A	7.2.10	
UnsignedProperties	M	6.2.2	See Section 6
UnsignedSignatureProperties	M	6.2.5	See Section 6
CounterSignature	N/A	7.2.4	

6. Qualified BDOC signatures

BDOC Basic Profile does not contain necessary elements to verify whether the signer's certificate was valid at the (claimed) time of signing. XAdES Basic Profile form may be used in internal systems where there are other (undocumented) means for dealing with the certificate validity issue.

The validation data shall be obtained as soon as possible after creation of the XAdES-BES signature. Two scenarios are applicable here:

- in desktop environment – the signer's application itself shall obtain validation data and optionally timestamp as soon as possible after signature creation;
- in web environment – the server-side application shall obtain validation data and optionally timestamp as soon as possible after receiving signature from the user.

As the signature creation is an off-line act and timing of thereof cannot be identified in trusted manner, this specification relies on notion that "signature creation time" shall be derived from the time information of validation and/or time-stamping services. In other words, signature shall not be considered *complete* or *valid* without validation data from external services.

This BDOC specification defines two methods for creating electronic signatures equal to "handwritten signature". Both profiles are compliant to XAdES LT-Level (see XAdES BP[8] section 8) requirements and are providing means for including certificate validity and time information with the signature:

- **time-marking**. In this scenario, the OCSP responder shall follow specific service requirements described in section 6.1. In this case, additional time-stamp service is not required
- **time-stamping**. This shall be used in case OCSP is not replacing need for additional trusted time-stamps from external Time-Stamping Authority. Refer to XAdES[1] clause 4.4.3.1.

Software implementations complying with current BDOC specification must support both abovementioned methods.

Both supported forms are using XAdES elements from "T" and "L" blocks.

Element		***XAdES clause***	***Comment***
SignatureTimeStamp		7.3	See paragraphs 6.1 and 6.2 below
CompleteCertificateRefs	N/A	7.4.1	
CompleteRevocationRefs	N/A	7.4.2	
AttributeCertificateRefs	N/A	7.4.3	
AttributeRevocationRefs	N/A	7.4.4	
SigAndRefsTimeStamp	N/A	7.5.1	
RefsOnlyTimeStamp	N/A	7.5.2	
CertificateValues	M	7.6.1	`EncapsulatedX509Certificate` is supported only. Must contain signer's CA certificate. In case OCSP response does not contain responder's certificate, it shall be present here. When using time-stamping (see 6.2), the TSA certificate shall also be included here in case it is not present in the time-stamp.
RevocationValues	M	7.6.2	`OCSPValues` and `EncapsulatedOCSPValue` elements are supported only. The OCSP response must have status "good" only.
AttrAuthoritiesCertValues	N/A	7.6.3	
AttributeRevocationValues	N/A	7.6.4	
xadesv141: TimeStampValidationData		8.1	See clause 7.2 below
xadesv141:ArchiveTimeStamp		8.2	See paragraph 7 below
UnsignedDataObjectProperties	N/A	6.2.6	

6.1. BDOC with time-marks

As the XAdES specification defines, “a time-mark provided by a Trusted Service would have similar effect to the time-stamp property but in this case no property is added to the electronic signature as it is the responsibility of the TSP to provide evidence of a time mark when required to do so.”

This BDOC specification defines a mechanism for time-marking through using OCSP [4] protocol. Immediately after signature creation, the digital signing application shall obtain a certificate validity confirmation using OCSP protocol. Hash value of the binary value of the signature along with hash algorithm identifier must be present in the “nonce” field of the OCSP request. The OCSP responder shall return this “nonce” value within the signed response. The “nonce” value must be the DER-encoding of the following ASN.1 data structure:

```
TBSDocumentDigest ::= SEQUENCE {
      algorithm AlgorithmIdentifier,
      digest OCTET STRING
}
```

The element `digest` is a hash value of the binary value of the signature and the element `algorithm` determines used hash algorithm defined in RFC 5280[12] clause 4.1.1.2. This mechanism solves the complexity of time-stamping and certificate validity relations in one step by combining these two services. The OCSP response described above shall be treated as a validity confirmation telling “at the time I saw this signature, corresponding certificate was valid”, digitally signed by the service. As a result no additional timestamp is required and element `SignatureTimeStamp` is not used.

OCSP service shall be “real-time” reflecting the freshest possible evidence about validity of the certificate. This means that difference between `thisUpdate` and `producedAt` values in OCSP response shall be measurable in seconds. The service shall follow principles documented in ETSI standard “Policy requirements for time-stamping authorities” [7].

The value of `producedAt` in the OCSP response shall be treated as a time of signature creation.

6.2. BDOC with time-stamps

BDOC profile with time-stamps is used in case OCSP service does not correspond to requirements described in paragraph 6.1. In this case, additional time-stamps are required in order to fix time of certificate validity information.

This is accomplished by including `SignatureTimeStamp` element to the signature structure. The time-stamp is obtained as soon as possible after signature creation.

`SignatureTimeStamp` is defined in clause 7.3 of XAdES[1]. The element is type of `XAdESTimeStampType` defined in clause 7.1.4.3. This BDOC specification further profiles the element as follows:

- Only IETF-style (RCF3161) time-stamps are supported i.e. `EncapsulatedTimeStamp` is supported only
- Attribute `Id` is mandatory

Time value in the element `SignatureTimeStamp` shall be treated as a time of signature creation.

7. Mechanisms for long-time validity

Qualified BDOC signature specified in last section is secure enough provided that cryptographic algorithms used are unbreakable, key lengths are sufficient and private keys of CSP (the CA and OCSP key) remain under control of the service provider.

Fast advances in computing suggest that key lengths and algorithms used today may not be secure enough in the future. There is also always a (theoretical) possibility that private key of some PKI service can get corrupted.

Additional measures are needed to protect electronic signatures from threats like those. This document describes two mechanisms for maintaining long-time validity of electronic signatures:

- **Logging:** service providing external evidence of certificate validity at the time of signing creates and maintains log containing issued responses
- **Archive time-stamp:** the whole material of the signature is periodically re-time-stamped

The first option does not require any end-user activity or additional functionality from BDOC-compliant system and therefore is preferred method. From the other hand the logging puts additional requirements to the service provider which may not be followed. In order to fully secure end-user and give him some additional independence of service provider, the archive time-stamp mechanism shall also be supported.

7.1. Logging

This mechanism builds on purpose of preserving evidence that "the signer's certificate was valid at the time of signing" for a long time.

Depending on the Qualified BDOC method used, corresponding services must create log of all issued responses:

- in case of time-marking (section 6.1), the OCSP service
- in case of time-stamping (section 6.2), the time-stamping service

The log entry shall be created **before** issuing a response. If the log entry creation fails, error response shall be produced by the service. This principle ensures existence of the every response given.

The service provider shall provide a public interface for verification of existence of certain response in the log.

For further fortification of the logging mechanism, additional security measures may be implemented:

- Cryptographic linking: every log record is dependent on the previous. This can be accomplished by creating a hash chain making every log record dependent on all others. This prevents from threats of deleting or interjecting falsified log records.
- Periodical publication of latest log record in printed media. This mechanism provides for a kind of non-repudiation property for the service provider – it takes away all possibilities to forge the log after publishing as the published log record represents the whole log. Of course, it is usable only when cryptographic linking is used.

Extreme care shall be taken on proper maintenance and back-up of the logs.

Described method does not protect BDOC signatures for loss of collision resistance of hash algorithm used for creation of the signature.

7.2. Archive time-stamp

This mechanism builds on notion "let's secure what may be weak". Successive time-stamps protect the whole material against vulnerable hashing algorithms or the breaking of the cryptographic material or algorithms.

It should be noted, that time-stamping is commonly a user-initiated act. In case digitally signed files are all scattered over user's computers (or even on external carriers), it might be tricky (or even impossible) to make users to time-stamp existing documents at proper time. However, re-time-stamping may be usable in case digitally signed files are stored in some central repository.

The BDOC archive time-stamping profile follows clause 8.2 of XAdES[1]. "Not distributed case" is supported only as defined in 8.2.1 of XAdES[1]. Element `XAdESv141:TimeStampValidationData` (clause 8.1) shall be used in case verification data for validation of the archive time-stamp is not yet present in the other places of BDOC signature. Like with `SignatureTimeStamp` element, only `EncapsulatedTimeStamp` is supported and element `Id` is mandatory.

8. Container Format

This section describes container format for packaging signed original files and signatures into one data unit (file). In other words, it defines what “digitally signed file” is.

The BDOC file format is based on ASiC[9] standard which is in turn profiled by ASiC BP[10]. The latter foresees ODF-type packaging specified in OpenDocument[5] standard of OASIS.

BDOC packaging is a ASiC-E XAdES type (see [10], clause 8.3) ZIP container with the following requirements followed:

1. **mimetype file**. The file “mimetype” shall be present in uncompressed form and formed as described in clause A.1 of the ASiC[9] standard. The content must be:

   ```
   application/vnd.etsi.asic-e+zip
   ```

2. **manifest file**. The file “manifest.xml” shall be present in directory META-INF/ and contain list of all directories and files with their types present in the container as described in section 3.2 of the OpenDocument[5] standard. The list shall not include file “mimetype” and files in the “META-INF/” directory (i.e. signature files).

Root element shall be the same type as in the “mimetype” file.

Digital signatures are stored in the META-INF/ folder, usually one file per signature. The names of these files shall contain the string "signatures". Root element of each signature file shall be `<asic:XAdESSignatures>`. Case where one signature file contains multiple signatures shall be supported.

As a rule, all files in BDOC container are signed except file “mimetype” and files in the META-INF directory. However, signed objects are explicitly referred by `<Reference>` elements in the signatures and therefore BDOC-compatible applications shall display list of signed files based on that. All signatures in a single BDOC container must refer to the same set of objects.

The file extension of BDOC file format is “.bdoc”, conforming applications may also support extensions “.asice” and “.sce”. MIME-type is “application/vnd.etsi.asic-e+zip”.

Appendix 1: Sample BDOC

The following sample BDOC file contains single embedded data file, one signature and has created with time-mark.

1. BDOC file structure

```
document.doc
mimetype
META-INF/manifest.xml
META-INF/signatures1.xml
```

2. Content of file “mimetype”

```
application/vnd.etsi.asic-e+zip
```

3. Content of file “META-INF/manifest.xml”

```
<?xml version="1.0" encoding="utf-8"?>
<manifest:manifest
xmlns:manifest="urn:oasis:names:tc:opendocument:xmlns:manifest:1.0">
<manifest:file-entry manifest:media-type="application/vnd.etsi.asic-e+zip"
manifest:full-path="/" />
<manifest:file-entry manifest:media-type="application/msword"
       manifest:full-path="document.doc" />
</manifest:manifest>
```

4. Content of file “META-INF/signatures1.xml”

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<asic:XAdESSignatures xmlns:asic="http://uri.etsi.org/02918/v1.2.1#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xades="http://uri.etsi.org/01903/v1.3.2#">

  <ds:Signature Id="S0">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2006/12/xml-
c14n11"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#rsa-sha224"/>
      <ds:Reference Id="S0-RefId0" URI="document.doc">
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
        <ds:DigestValue>5UyKB9ht94y6CZNvLdO1C7Z3MXaYc2Qol3Dt3Qp4Ajg=
</ds:DigestValue>
      </ds:Reference>
      <ds:Reference Id="S0-RefId1"
Type="http://uri.etsi.org/01903#SignedProperties" URI="#S0-SignedProperties">
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
     <ds:DigestValue>YGDmd4GaWLgV4/hrEVV6/DvQ6uLhfnTSI0CQJX612KM=
</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue Id="S0-SIG">
YQs06u9ekMnZd2Jy+Won5VK0kIC9y5e2JPfraUItZOqwxd4rc4g3fiUnDkrf
iHIdD2xOGyszCZA/JAicqDPiFkmXbjkgpYYF8gY3NB/xFwoKv/zaWu7HEi+T
```

```
eq/OoSDlXVGi0H++27nI3xAl7P7Iz84xaji1aquZQVl5iOtWD8k=
    </ds:SignatureValue>
     <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>
MIIEnDCCA4SgAwIBAgIQfybdp3nKOMhPqk9YDxgaTTANBgkqhkiG9w0BAQU...
x3CqdYNWwQhU2bMirW4=
       </ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
    <ds:Object>
      <xades:QualifyingProperties Target="#S0">
        <xades:SignedProperties Id="S0-SignedProperties">
          <xades:SignedSignatureProperties>
            <xades:SigningTime>2012-12-09T15:49:32Z</xades:SigningTime>
            <xades:SigningCertificate>
              <xades:Cert>
                <xades:CertDigest>
                  <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                  <ds:DigestValue>z/CsSIOu/w41P63VzQEXRkxsT/oht2ggvA6rMxDQvoA=
</ds:DigestValue>
                </xades:CertDigest>
                <xades:IssuerSerial>
                  <ds:X509IssuerName>emailAddress=pki@sk.ee,CN=TEST of ESTEID-
SK 2011,O=AS Sertifitseerimiskeskus,C=EE</ds:X509IssuerName>
<ds:X509SerialNumber>169013758426626343561532977746185558605</ds:X509SerialNumb
er>
                </xades:IssuerSerial>
              </xades:Cert>
            </xades:SigningCertificate>
            <xades:SignaturePolicyIdentifier>
              <xades:SignaturePolicyId>
                <xades:SigPolicyId>
                  <xades:Identifier
Qualifier="OIDAsURN">urn:oid:1.3.6.1.4.1.10015.1000.3.2.3</xades:Identifier>
                </xades:SigPolicyId>
                <xades:SigPolicyHash>
                  <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                  <ds:DigestValue>*** HERE IS THE HASH VALUE OF THE CURRENT
DOCUMENT IN BASE64 ENCODING***</ds:DigestValue>
                </xades:SigPolicyHash>
                <xades:SigPolicyQualifiers>
                  <xades:SigPolicyQualifier>
                    <xades:SPURI>https://www.sk.ee/repository/bdoc-
spec212.pdf</xades:SPURI>
                  </xades:SigPolicyQualifier>

                </xades:SigPolicyQualifiers>
              </xades:SignaturePolicyId>
            </xades:SignaturePolicyIdentifier>
            <xades:SignatureProductionPlace>
             <xades:City>Tallinn</xades:City>
             <xades:StateOrProvince>Harju</xades:StateOrProvince>
             <xades:PostalCode>10122</xades:PostalCode>
             <xades:CountryName>Estonia</xades:CountryName>
            </xades:SignatureProductionPlace>
            <xades:SignerRole>
             <xades:ClaimedRoles>
              <xades:ClaimedRole>Agreed</xades:ClaimedRole>
             </xades:ClaimedRoles>
           </xades:SignerRole>
```

```
          </xades:SignedSignatureProperties>
          <xades:SignedDataObjectProperties>
            <xades:DataObjectFormat ObjectReference="#S0-RefId0">
              <xades:MimeType>application/msword</xades:MimeType>
            </xades:DataObjectFormat>
          </xades:SignedDataObjectProperties>
        </xades:SignedProperties>
   <xades:UnsignedProperties>
    <xades:UnsignedSignatureProperties>
     <xades:CertificateValues>
      <xades:EncapsulatedX509Certificate Id="S0-CA_CERT">
MIIDPDCCAiSgAwIBAgIEQi2iwTANBgkqhkiG9w0BAQUFADB8MRgwFgYJKoZIhvcN
       ...
       EWyMVkNnZooWHIjLpNucQA==
      </xades:EncapsulatedX509Certificate>
      <xades:EncapsulatedX509Certificate Id="S0-RESPONDER_CERT">
MIIEITCCAwmgAwIBAgIBDDANBgkqhkiG9w0BAQUFADCBgDELMAkGA1UEBhMCSUUx
       ...
       EWyMVkNnZooWHIjLpNucQA==
      </xades:EncapsulatedX509Certificate>
     </xades:CertificateValues>
     <xades:RevocationValues>
      <xades:OCSPValues>
       <xades:EncapsulatedOCSPValue Id="N0">
MIIBtgoBAKCCAa8wggGrBgkrBgEFBQcwAQEEggGcMIIBmDCCAQGhcTBvMQswCQYD
        ...
        knf8XDhdklVD0w==
        </xades:EncapsulatedOCSPValue>
       </xades:OCSPValues>
      </xades:RevocationValues>
     </xades:UnsignedSignatureProperties>
    </xades:UnsignedProperties>
   </xades:QualifyingProperties>
  </xades:Object>
 </xades:Signature>
</asic:XAdESSignatures>
```

Appendix 2: BDOC Signature Profiles

The following figure illustrates use of XAdES elements within different BDOC-defined signature profiles.