

Advanced Electronic Signature Profiles of the OASIS Digital Signature Service Version 1.0

5 Committee Specification

6 13 February 2007

7	Specification URIs:
8 9	This Version: http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-AdES-cs-v1.0-r1.html
10	http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-AdES-cs-v1.0-r1.pdf
11 12	Latest Version: http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-AdES-cs-v1.0-r1.html
13	http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-AdES-cs-v1.0-r1.pdf
14 15	Technical Committee: OASIS Digital Signature Services TC
16 17 18	Chair(s): Nick Pope, Thales eSecurity Juan Carlos Cruellas, Centre d'aplicacions avançades d'Internet (UPC)
19 20	Editor(s): Juan Carlos Cruellas, Centre d'aplicacions avançades d'Internet (UPC)
21 22	Related work: This specification is related to:
23	 oasis-dss-core-spec-cs-v1.0-r1
24 25 26 27 28 29	Abstract: This document defines one abstract profile of the OASIS DSS protocols for the purpose of creating and verifying XML or CMS based Advanced Electronic Signatures. It also defines two concrete sub-profiles: one for creating and verifying XML Advanced Electronic Signatures and the other for creating and verifying CMS based Advanced Electronic Signatures.
30 31 32 33 34	Status: This document was last revised or approved by the OASIS Digital Signature Services TC on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

oasis-dss-profiles-AdES-spec-cs-v1.0-r1

13 February 2007

35 36 37 38	Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at http://www.oasis-open.org/committees/dss.
39 40 41 42	For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/dss/ipr.php.
43 44	The non-normative errata page for this specification is located at http://www.oasis-open.org/committees/dss.

Notices

45

- 46 OASIS takes no position regarding the validity or scope of any intellectual property or other
- 47 rights that might be claimed to pertain to the implementation or use of the technology
- 48 described in this document or the extent to which any license under such rights might or might
- 49 not be available; neither does it represent that it has made any effort to identify any such
- rights. Information on OASIS's procedures with respect to rights in OASIS specifications can
- 51 be found at the OASIS website. Copies of claims of rights made available for publication and
- any assurances of licenses to be made available, or the result of an attempt made to obtain a
- 53 general license or permission for the use of such proprietary rights by implementors or users
- of this specification, can be obtained from the OASIS Executive Director.
- 55 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
- 56 applications, or other proprietary rights which may cover technology that may be required to
- 57 implement this specification. Please address the information to the OASIS Executive Director.
- 58 Copyright © OASIS® 1993–2007. All Rights Reserved. OASIS trademark, IPR and other
- 59 policies apply.
- 60 This document and translations of it may be copied and furnished to others, and derivative
- 61 works that comment on or otherwise explain it or assist in its implementation may be
- 62 prepared, copied, published and distributed, in whole or in part, without restriction of any kind,
- 63 provided that the above copyright notice and this paragraph are included on all such copies
- 64 and derivative works. However, this document itself may not be modified in any way, such as
- 65 by removing the copyright notice or references to OASIS, except as needed for the purpose
- 66 of developing OASIS specifications, in which case the procedures for copyrights defined in
- 67 the OASIS Intellectual Property Rights document must be followed, or as required to translate
- it into languages other than English.
- 69 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
- 70 successors or assigns.
- 71 This document and the information contained herein is provided on an "AS IS" basis and
- 72 OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT
- 73 LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
- 74 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY
- 75 OR FITNESS FOR A PARTICULAR PURPOSE.
- 76 The names "OASIS" are trademarks of OASIS, the owner and developer of this specification,
- 77 and should be used only to refer to the organization and its official outputs. OASIS welcomes
- 78 reference to, and implementation and use of, specifications, while reserving the right to
- 79 enforce its marks against misleading uses. Please see http://www.oasis-
- open.org/who/trademark.php for above guidance.

Table of Contents

81

82	1 Introduction	7
83 84	1.1 Terminology	7
85	1.3 Non-Normative References	
86	1.4 Namespaces	
87	2 Overview	9
88	3 Advanced Electronic Signature abstract profile	10
89	3.1 Overview	10
90	3.2 Profile Features	
91	3.2.1 Scope	
92	3.2.2 Relationship To Other Profiles	
93	3.2.3 Signature Object	
94	3.3 Profile of Signing Protocol	
95	3.3.1 Element <signrequest></signrequest>	
96	3.3.1.1 Element <optionalinputs></optionalinputs>	
97	3.3.1.1.1 New Optional Inputs	
98	3.3.1.1.1.1 Optional Input <signatureform></signatureform>	
99	3.3.1.1.2 Optional Inputs already defined in the Core	
100 101	3.3.1.1.2.1 Optional Input <signaturetype></signaturetype>	
101	3.3.1.1.2.2 Optional Inputs Chairnedidentity and KeySelectors	
102	3.3.1.1.2.3 Optional input < signled rioperties >	
103	3.3.1.1.2.3.2 Requesting CommitmentTypeIndication	
105	3.3.1.1.2.3.3 Requesting SignatureProductionPlace	14
106	3.3.1.1.2.3.4 Requesting SignerRole	15
107	3.3.1.1.2.3.5 Requesting AllDataObjectsTimeStamp	
108	3.3.1.1.2.3.6 Requesting DataObjectFormat	
109	3.3.2 Element <signresponse></signresponse>	
110	3.3.2.1 Element <signatureobject></signatureobject>	
111	3.3.2.2 Optional Outputs	
112	3.4 Profile of Verifying Protocol	
113	3.4.1 Element <verifyrequest></verifyrequest>	16
114	3.4.1.1 Attribute Profile	
115	3.4.1.2 Element <signatureobject></signatureobject>	
116	3.4.1.3 Element < Optional Inputs>	
117	3.4.1.3.1 Element <returnupdatedsignature></returnupdatedsignature>	16
118	3.5 Element <verifyresponse></verifyresponse>	17
119	3.5.1.1 Element < Optional Outputs>	
120	3.5.1.1.1 Optional Output <updatedsignature></updatedsignature>	17
121	4 XML Advanced Electronic Signatures concrete Profile	18
122	4.1 Overview	18
123	4.2 Profile features	
124	4.2.1 Identifier	
125	4.2.2 Scope	19
126	4.2.3 Relationship To Other Profiles	
127	4.2.4 Signature Object	
128	4.2.5 Transport Binding	

129	4.2.6 Security Binding	19
130	4.3 Profile of Signing Protocol	19
131	4.3.1 Attribute Profile	19
132	4.3.2 Element <signrequest></signrequest>	20
133	4.3.2.1 Element <optionalinputs></optionalinputs>	20
134	4.3.2.1.1 New Optional Inputs	
135	4.3.2.1.1.1 Element <signatureform></signatureform>	
136	4.3.2.1.2 Optional Inputs already defined in the Core	
137	4.3.2.1.2.1 Optional Input <signaturetype></signaturetype>	
138	4.3.2.1.2.2 Optional inputs < ClaimedIdentity> and <keyselector></keyselector>	
139	4.3.2.1.2.3 Optional Input <signedproperties></signedproperties>	
140	4.3.2.1.2.3.1 Requesting SigningTime	
141	4.3.2.1.2.3.2 Requesting CommitmentTypeIndication	
142	4.3.2.1.2.3.3 Requesting SignatureProductionPlace	
143	4.3.2.1.2.3.4 Requesting SignerRole	
143 144		
	4.3.2.1.2.3.5 Requesting AllDataObjectTimeStamp	
145	4.3.2.1.2.3.6 Requesting DataObjectFormat	21
146	4.3.2.1.2.3.7 Requesting <xades:individualdataobjecttimestamp></xades:individualdataobjecttimestamp>	
147	4.3.3 Element <signresponse></signresponse>	
148	4.3.3.1 Element <signatureobject></signatureobject>	
149	4.4 Profile of Verifying Protocol	
150	4.4.1 Element <verifyrequest></verifyrequest>	
151	4.4.1.1 Attribute Profile	
152	4.4.1.2 Element <signatureobject></signatureobject>	
153	4.4.1.3 Element <optionalinputs></optionalinputs>	
154	4.4.1.3.1 Optional Output <returnupdatedsignature></returnupdatedsignature>	23
155	4.4.2 Element <verifyresponse></verifyresponse>	
156	4.4.2.1 Element <optionaloutputs></optionaloutputs>	23
157	4.4.2.1.1 Optional Output <updatedsignature></updatedsignature>	23
158	4.5 Profile Bindings	23
159	4.5.1 Transport Bindings	23
160	4.5.2 Security Bindings	23
161	4.5.2.1 Security Requirements	23
162	4.5.2.2 TLS X.509 Mutual Authentication	23
400		
163	5 CMS-based Advanced Electronic Signature profile	24
164	5.1 Overview	24
165	5.2 Profile features	
166	5.2.1 Identifier	_
167	5.2.2 Scope	
168	5.2.3 Relationship To Other Profiles	
169	5.2.4 Signature Object	
170	5.2.5 Transport Binding	
171	5.2.6 Security Binding	
172	5.3 Profile of Signing Protocol	
173	5.3.1 Element <signrequest></signrequest>	
	· · · · · · · · · · · · · · · · · · ·	
174 175		
175	5.3.1.2 Element < Optional Inputs >	
176	5.3.1.2.1 New Optional Inputs	
177	5.3.1.2.1.1 Element <signatureform></signatureform>	
178	5.3.1.2.2 Optional Inputs already defined in the Core	
179	5.3.1.2.2.1 Element <signaturetype></signaturetype>	26
180	5.3.1.2.2.2 Optional inputs < ClaimedIdentity> / <keyselector></keyselector>	
181	5.3.1.2.2.3 Element <signedproperties></signedproperties>	26

182	5.3.1.2.2.3.1 Requesting signing-time	. 26
183	5.3.1.2.2.3.2 Requesting commitment-type-indication	. 26
184	5.3.1.2.2.3.3 Requesting signer-location	. 26
185	5.3.1.2.2.3.4 Requesting signer-attributes	. 27
186	5.3.1.2.2.3.5 Requesting content-time-stamp	. 27
187	5.3.1.2.2.3.6 Requesting content-hints	. 27
188	5.3.2 Element <signresponse></signresponse>	. 27
189	5.3.2.1 Element <signatureobject></signatureobject>	. 27
190	5.4 Profile of Verifying Protocol	. 27
191	5.4.1 Element <verifyrequest></verifyrequest>	. 27
192	5.4.1.1 Attribute Profile	
193	5.4.1.2 Element <optionalinputs></optionalinputs>	. 28
194	5.4.1.2.1 Element <returnupdatedsignature></returnupdatedsignature>	. 28
195	5.4.1.3 Element <signatureobject></signatureobject>	. 28
196	5.4.2 Element <verifyresponse></verifyresponse>	
197	5.4.2.1 Element <optionaloutputs></optionaloutputs>	. 28
198	5.4.2.1.1 Element < Updated Signature >	
199	5.5 Profile Bindings	
200	5.5.1 Transport Bindings	
201	5.5.2 Security Bindings	
202	5.5.2.1 Security Requirements	
203	5.5.2.2 TLS X.509 Mutual Authentication	. 28
204	6 XML timestamps in XAdES signatures	. 29
205	6.1 Generation and inclusion of XML timestamps	. 29
206	6.1.1 Profile for XAdES timestamp containers	
207	6.1.2 XML timestamp within xades:IndividualDataObjectsTimeStamp	
208	6.1.3 XML timestamp within xades:AllDataObjectsTimeStamp	
209	6.1.4 XML timestamp within xades:SigAndRefsTimeStamp	
210	6.1.5 XML timestamp within xades:RefsOnlyTimeStamp	
211	6.1.6 XML timestamp within xades:ArchiveTimeStamp	
212	6.2 Verification of XML timestamps	
213	6.2.1 Verification of of xades:IndividualIDataObjectsTimeStamp including a XML	
214	timestamp	. 32
215	6.2.2 Verification of xades:AllDataObjectsTimeStamp including a XML timestamp	. 32
216	6.2.3 Verification of xades:SigAndRefsTimeStamp including a XML timestamp	
217	6.2.4 Verification of xades:RefsOnlyTimeStamp including a XML timestamp	. 34
218	6.2.5 Verification of xades:ArchiveTimeStamp including a XML timestamp	. 34
219	7 Identifiers defined in this specification	. 36
220	7.1 Predefined advanced electronic signature forms identifiers	
221	7.2 Result Identifiers	
222	A. Acknowledgements	. 38

1 Introduction

224

231

240

242

244

249

252

255

258

260

264

- 225 The DSS signing and verifying protocols are defined in [DSSCore]. As defined in that
- document, the DSS protocols have a fair degree of flexibility and extensibility. This document
- defines an abstract profile for the use of the DSS protocols for creating and verifying XML and
- 228 CMS-based Advanced Electronic Signatures as defined in [XAdES] and [CAdES]. This
- document also defines two concrete profiles derived from the abstract one: one for creating
- and verifying XAdES signatures and the other for creating and verifying CAdES signatures.

1.1 Terminology

- The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
- 233 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be
- interpreted as described in IETF RFC 2119 [RFC 2119]. These keywords are capitalized
- 235 when used to unambiguously specify requirements over protocol features and behavior that
- 236 affect the interoperability and security of implementations. When these words are not
- capitalized, they are meant in their natural-language sense.
- 238 This specification uses the following typographical conventions in text: <ns:Element>,
- 239 Attribute, Datatype, OtherCode.

1.2 Normative References

- 241 [AdES-XSD] J. C. Cruellas et al. AdES Profile Schema, OASIS, February 2007.
- 243 [CAdES] CMS Advanced Electronic Signatures. ETSI TS 101 733, January 2007.
- 245 [Core-XSD] S. Drees et al. DSS Schema. OASIS, February 2007).
- 246247 [DSSCore] S. Drees et al. Digital Signature Service Core Protocols and Elements.
- 248 OASIS, February 2007.
- 250 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.
- 253 **[RFC 2634]** . Hoffman (ed.). Enhanced Security Services for S/MIME http://www.ietf.org/rfc/rfc2634.txt, , IETF RFC 2634 June 1999
- 256 **[RFC 3852]** R. Housley. Cryptographic Message Syntax (CMS), IETF RFC 3852, July 257 2004.
- 259 [XAdES] Advanced Electronic Signatures. ETSI TS 101 733. March 2006.
- [XML-ns] T. Bray, D. Hollander, A. Layman. Namespaces in XML.
 http://www.w3.org/TR/1999/REC-xml-names-19990114, W3C Recommendation, January
- 263 1999.

- 265 **[XMLSig]** D. Eastlake et al. *XML-Signature Syntax and Processing.*
- 266 http://www.w3.org/TR/1999/REC-xml-names-19990114, W3C Recommendation, February
- 267 2002.

268

1.3 Non-Normative References

269 1.4 Namespaces

- 270 The structures described in this specification are contained in the schema file [AdES-XSD]. All
- 271 schema listings in the current document are excerpts from the schema file. In the case of a
- disagreement between the schema file and this document, the schema file takes precedence.
- 273 This schema is associated with the following XML namespace:
- 274 urn:oasis:names:tc:dss:1.0:profiles:AdES:schema#
- 275 Conventional XML namespace prefixes are used in this document:
- 276 o The prefix dss: (or no prefix) stands for the DSS core namespace [Core-XSD].
- 277 o The prefix ds: stands for the W3C XML Signature namespace [XMLSig].
- o The prefix xades: stands for ETSI XML Advanced Electronic Signatures (XAdES) document [XAdES].
- 280 Applications MAY use different namespace prefixes, and MAY use whatever namespace
- defaulting/scoping conventions they desire, as long as they are compliant with the
- Namespaces in XML specification [XML-ns].

Page 8 of 38

2 Overview 283 284 This document defines three profiles of the protocols specified in: "Digital Signature Services Core Protocol and Elements" [DSSCore]. 285 286 The first one is an abstract profile defining messages for supporting the lifecycle of advanced 287 electronic signatures. Both, XML and CMS-based advanced electronic signatures are 288 supported by this profile. 289 One concrete profile, derived from the aforementioned abstract profile, gives support to the 290 lifecycle of XML advanced electronic signatures as specified in [XAdES]. A second concrete profile, also derived from the abstract one, gives support to the lifecycle of 291 292 CMS-based advanced electronic signatures as specified in [CAdES]. 293 Implementations should implement one of the concrete profiles (or both) in order to request 294 generation or validation of advanced electronic signatures in one of the two formats (or both).

3 Advanced Electronic Signature abstract profile

_			_							
- 2	1	(•	10	PI	/ i		AAA	i
) _ I			, ,	<i>,</i> –		•	_	vv	

295

296

297

298

299 300

301 302

303

304 305

306

307

308 309

310

311

312 313

314

315

316

317

318 319

320

321 322

323

324

325 326

327

328

329

330 331

332

333

This abstract profile supports operations within each phase of the lifecycle of two types of advanced electronic signature:

- o XML encoded signatures based on [XMLSig] such as specified in [XAdES].
- o Binary encoded signatures based on [RFC 3852] such as specified in [CAdES].

Henceforward, the document will use the term advanced signature when dealing with issues that affect to both types of signatures. The document will use XAdES or CAdES signatures when dealing with issues that affect one or the other but not both of them.

For the generation of advanced signatures, the following operations apply:

- SignRequest. This operation supports requests for:
 - Generating predefined advanced signature forms as defined in [XAdES] and [CAdES].
 - Generating XML signatures incorporating specific signed/unsigned properties whose combination does not fit any predefined XAdES signature form. In such cases, the form MUST have been defined in a proprietary specification and MUST be identified by one URI.
 - Generating CMS signatures incorporating specific signed/unsigned attributes whose combination does not fit any predefined [CAdES] signature form. In such cases, the form MUST have been defined in a proprietary specification and MUST be identified by one URI.
- SignResponse. This operation supports delivery of:
 - Predefined advanced signature forms as defined in [XAdES] and [CAdES].
 - XML signatures with specific properties whose combination does not fit any predefined XAdES signature form. In such cases, the form MUST have been defined in some other specification and MUST be identified by one URI.
 - CMS signatures incorporating specific signed attributes whose combination does not fit any predefined [CAdES] signature form. In such cases, the form MUST have been defined in some other specification and MUST be identified by one URI.

For advanced signature verification (and updating) the following operations apply:

- o VerifyRequest. This operation supports requests for:
 - Verifying a predefined advanced signature form.
 - Verifying XML signatures incorporating specific properties whose combination does not fit any predefined XAdES signature form.
 - Verifying any of the signatures mentioned above PLUS updating them by addition of additional properties (time-stamps, validation data, etc) leading to a predefined XAdES form.

- 334 Verifying CMS signatures incorporating specific attributes whose combination 335 does not fit any predefined [CAdES] signature form. 336 Verifying any of the signatures mentioned above PLUS updating them by addition of additional attributes (time-stamps, validation data, etc) leading to a 337 338 predefined [CAdES] form. 339 Verifying a long-term advanced signature in a certain point of time. VerifyResponse. This operation supports delivery of: 340 341 Advanced signature verification result of signatures mentioned above.
 - Advanced signature verification result PLUS the updated signatures as
 - Advanced signature verification result PLUS the updated signatures as requested.
- The material for each operation will clearly indicate the lifecycle phase it pertains to.

3.2 Profile Features

346 **3.2.1 Scope**

342 343

345

348

359

360

361

362

347 This document profiles the DSS signing and verifying protocols defined in [DSSCore].

3.2.2 Relationship To Other Profiles

- The profile in this document is based on the [DSSCore]. The profile in this document may not
- 350 be directly implemented. It is further profiled by the two concrete profiles also defined in
- 351 sections 4 and 5.

352 3.2.3 Signature Object

- 353 This profile supports the creation and verification of advanced signatures as defined in
- 354 [XAdES] and [CAdES].
- 355 This profile also supports update of advanced signatures by addition of unsigned properties
- 356 (time-stamps and different types of validation data), as specified in [XAdES] and [CAdES].

357 3.3 Profile of Signing Protocol

- 358 The present profile allows requesting:
 - Predefined forms of advanced electronic signatures as defined in [XAdES] and [CAdES].
 - Other forms of signatures based in [XMLSig] or [RFC 3852] defined in other specifications,
- In both cases, the specific requested form will be identified by an URI.
- According to this profile, the following predefined advanced signature forms defined in [XAdES] and [CAdES] MAY be requested (those forms whose name begin by XAdES- are forms names for XAdES signatures; those ones whose name begin by CAdES are names for CAdES signatures):
- o CAdES-BES and XAdES-BES. In this form, the signing certificate is secured by the signature itself.

- o CAdES-EPES and XAdES-EPES. This form incorporates an explicit identifier of the signature policy that will govern the signature generation and verification.
- o CAdES-ES-T and XAdES-T. This form incorporates a trusted time, by means of a time-stamp token or a time-mark.
- o CAdES-ES-C and XAdES-C.
- o CAdES-ES-X and XAdES-X.
- o CAdES-ES-X-L and XAdES-X-L.
- 377 o CAdES-ES-A and XAdES-A.
- 378 In addition, the present profile provides means for requesting incorporation in any of the
- 379 aforementioned forms any of the signed properties defined in [XAdES] and signed attributes
- 380 defined in [CAdES].
- 381 Other electronic signature forms based in [XMLSig] or [RFC 3852], defined elsewhere, MAY
- also be requested using the mechanisms defined in this profile.

383 3.3.1 Element <SignRequest>

384 This clause profiles the dss:SignRequest element.

385 3.3.1.1 Element < OptionalInputs>

386 3.3.1.1.1 New Optional Inputs

387 3.3.1.1.1.1 Optional Input <SignatureForm>

- 388 The form of signature required MAY be indicated using the following new optional input
- 389 <xs:element name="SignatureForm" type="xs:anyURI"/>
- 390 If not present the signature form SHALL be implied by the selected <SignaturePolicy> or
- 391 the signature policy applied by the server.
- 392 Section 7.1 of this abstract profile defines a set of URIs identifying the predefined advanced
- 393 electronic signature forms specified in [CAdES] and [XAdES].
- 394 Should other standard or proprietary specification define new signature forms and their
- 395 corresponding URIs, concrete sub-profiles of this abstract profile could be defined for giving
- 396 support to their verification and update.
- 397 Should a form identified by an URI, admit different properties combinations, the server will
- 398 produce a specific combination depending on its policy or configuration settings.

399 3.3.1.1.2 Optional Inputs already defined in the Core

- 400 None of the optional inputs specified in the [DSS Core] are precluded in this abstract profile. It
- 401 only constrains some of them and specifies additional optional inputs.

402 3.3.1.1.2.1 Optional Input <SignatureType>

403 This element is OPTIONAL. If present, <SignatureType> SHALL be either:

404 urn:ietf:rfc:3275

405 for requesting XML-based signatures, or

oasis-dss-profiles-AdES-spec-cs-v1.0-r1

13 February 2007

- 406 urn:ietf:rfc:3369
- for requesting CMS-based signatures, as defined in 7.1 of [DSS Core].
- 408 If not present the signature type SHALL be implied by the selected <SignaturePolicy> or
- 409 the signature policy applied by the server.

410 3.3.1.1.2.2 Optional inputs <ClaimedIdentity> and <KeySelector>

- As forms defined in [XAdES] and [CAdES] require that the signing certificate is protected by
- the signature, the server MUST gain access to that certificate.
- 413 <dss:ClaimedIdentity> or <dss:KeySelector> optional inputs MAY be present. If
- 414 they are not present, the server may use means not specified in this profile to identify the
- 415 signer's key and gain access to its certificate.

416 3.3.1.1.2.3 Optional Input <SignedProperties>

- The requester MAY request to the server the addition of optional signed properties using the
- 418 <dss:SignedProperties> element's <dss:Property> child profiled as indicated in
- 419 clauses below. First names correspond to the one given by XAdES to the signed properties.
- 420 Second ones correspond to the names given by CAdES to the signed attributes.
- 421 Signed properties that MAY be requested are:

XAdES	CAdES
SigningTime	signing-time
CommitmentTypeIndication	commitment-type-indication
SignerRole	signer-attributes
SignatureProductionPlace	signer-location
DataObjectFormat	content-hints
AllDataObjectsTimeStamp	content-time-stamp
IndividualDataObjectsTimeStamp	No equivalent signed attribute

- 422
- 423 Next sub-sections show how a client should request each of the aforementioned properties-
- 424 attributes. The type of signature requested (XAdES or CAdES) will determine whether a
- 425 XAdES property or a CAdES attribute is generated by the server.

426 3.3.1.1.2.3.1 Requesting SigningTime

427 Value for <Identifier> element:

428 urn:oasis:names:tc:dss:1.0:profiles:AdES:SigningTime

429 If the client does not request such property, the server still MAY generate and include this

430 property depending on its policy.

- 431 No content is required for Value element, since the actual contents of the property will be
- 432 generated by the server when required.

433 3.3.1.1.2.3.2 Requesting CommitmentTypeIndication

434 Value for <Identifier> element:

435 urn:oasis:names:tc:dss:1.0:profiles:AdES:CommitmentTypeIndication

- 436 If the client does not request such property, the server still MAY generate and include it with
- 437 values that depend on server's policy.
- The client MAY request the generation and inclusion of this signed property. In such cases
- 439 the <Value> element MUST have the following content:

```
440
      <xs:element name="RequestedCommitment">
441
          <xs:complexType>
442
              <xs:choice>
443
                  <xs:element ref="xades:CommitmentTypeIndication"/>
444
                  <xs:element name="BinaryValue" type="xs:base64Binary"/>
445
              </xs:choice>
446
          </xs:complexType>
447
      </xs:element>
```

- 449 signature.

455

- 450 Element <BinaryValue> will be present when requesting an ASN.1 signature. Its contents
- 451 MUST be the base64 encoding of commitment-type-indication ASN.1 attribute defined
- 452 in [CAdES], DER-encoded

453 3.3.1.1.2.3.3 Requesting SignatureProductionPlace

454 Value for <Identifier > element:

urn:oasis:names:tc:dss:1.0:profiles:AdES:SignatureProductionPlace

- The client MAY request a certain value for this property. Nevertheless, this value MAY be
- 457 ignored by the server depending on its own policy, and the property be set to another value.
- For requesting a value for this property, the <Value> element MUST have the following content:

```
460
      <xs:element name="RequestedSignatureProductionPlace">
461
          <xs:complexType>
462
              <xs:choice>
463
                  <xs:element ref="xades:SignatureProductionPlace"/>
464
                   <xs:element name="BinaryValue" type="xs:base64Binary"/>
465
              </xs:choice>
466
          </xs:complexType>
467
      </xs:element>
```

- 468 Element <xades: SignatureProductionPlace> will be present when requesting a XML
- 469 signature.
- 470 Element <BinaryValue> will be present when requesting an ASN.1 signature. Its contents
- 471 MUST be the base64 encoding of signerLocation ASN.1 attribute defined in [CAdES],
- 472 DER-encoded.

3.3.1.1.2.3.4 Requesting SignerRole

474 Value for <Identifier > element:

urn:oasis:names:tc:dss:1.0:profiles:AdES:SignerRole

When the client requests the generation and inclusion of this signed property the <Value> element MUST have the following content:

```
478
      <xs:element name="RequestedSignerRole">
479
          <xs:complexType>
480
              <xs:choice>
481
                  <xs:element ref="xades:SignerRole"/>
482
                  <xs:element name="BinaryValue" type="xs:base64Binary"/>
483
              </xs:choice>
484
          </xs:complexType>
485
      </xs:element>
```

- 486 Element <xades: SignerRole> will be present when requesting a XML signature.
- 487 Element <BinaryValue> will be present when requesting a ASN.1 signature. Its contents
- 488 MUST be the base64 encoding of signer-attributes ASN.1 attribute defined in [CAdES].
- 489 DER-encoded.

473

475

499

500

501

490 3.3.1.1.2.3.5 Requesting AllDataObjectsTimeStamp

- This element will be added for requesting the generation and inclusion of a time-stamp token
- on (all) the data object(s) to be signed.
- 493 Value for <Identifier> element:

494 urn:oasis:names:tc:dss:1.0:profiles:AdES:AllDataObjectsTimeStamp

- No content is required for <Value> element, since the actual contents of the property will be
- 496 generated by the server when required.

497 3.3.1.1.2.3.6 Requesting DataObjectFormat

498 Value for Identifier element:

urn:oasis:names:tc:dss:1.0:profiles:AdES:DataObjectFormat

When the client requests the generation and inclusion of this signed property the <Value> element MUST have the following content.

```
502
503
      <xs:element name="RequestedDocsFormat" type="DocsFormatType"/>
504
505
      <xs:complexType name="DocsFormatType">
506
          <xs:sequence>
507
              <xs:choice>
508
                   <xs:element name="DocFormat" type="DocFormatType"</pre>
509
      maxOccurs="unbounded"/>
510
                   <xs:element name="BinaryValue" type="xs:base64Binary"/>
511
               </xs:choice>
512
           </xs:sequence>
513
      </xs:complexType>
514
515
      <xs:complexType name="DocFormatType">
516
           <xs:complexContent>
517
              <xs:extension base="DocReferenceType">
518
                   <xs:sequence>
```

oasis-dss-profiles-AdES-spec-cs-v1.0-r1

519	<pre><xs:element ref="xades:DataObjectFormat"></xs:element></pre>
	Ab Clement Tel- Addeb Data Objection and 77
520	
521	
	\/ XB • EXCERSION>
522	
523	
323	

- 524 Elements < DocFormat > will be present when requesting an XML based signature.
- 525 Element <BinaryValue> will be present when requesting a CMS based signature. Its
- 526 contents MUST be the base64 encoding of content-hints ASN.1 attribute defined in [RFC
- 527 2634] DER-encoded.

528 3.3.2 Element < SignResponse >

529 This clause profiles the dss:SignResponse element.

530 3.3.2.1 Element < Signature Object>

531 This element SHALL NOT contain a dss:TimeStamp element as a child.

532 **3.3.2.2 Optional Outputs**

- None of the optional outputs specified in the [DSS Core] are neither precluded nor further
- profiled in this abstract profile.

535 3.4 Profile of Verifying Protocol

536 3.4.1 Element < VerifyRequest>

- This clause specifies the profile for the contents of the dss: VerifyRequest when used for:
- 538 o Requesting verification of advanced signatures.
- o Requesting verification of advanced signatures AND update of signatures to other predefined forms.

541 3.4.1.1 Attribute Profile

- The value for the Profile attribute, indicating the concrete sub-profile of this abstract profile,
- 543 MUST be present.

544 3.4.1.2 Element <SignatureObject>

This element SHALL NOT contain a dss:TimeStamp element as a child.

546 3.4.1.3 Element < OptionalInputs>

- None of the optional inputs specified in the [DSS Core] are precluded in this abstract profile. It
- only constrains some of them and specifies additional optional inputs.

549 3.4.1.3.1 Element < Return Updated Signature >

- 550 This element MUST be present when the client requests verification of a signature and
- update to a predefined form of advanced signature.
- 552 The Type attribute identifies the advanced signature form requested.

oasis-dss-profiles-AdES-spec-cs-v1.0-r1

13 February 2007

Copyright © OASIS Open 2007. All Rights Reserved.

Page 16 of 38

- Acceptable predefined values for this attribute are the URIs specified in table 1 corresponding
- to the following forms predefined in [CAdES] and [XAdES]: XAdES-T/CAdES -T, XAdES-
- 555 C/CAdES-C, XAdES-X/CAdES-X,XAdES-X-L/CAdES-X-L, XAdES-A/CAdES-A.
- 556 Should other standard or proprietary specification define new signature forms and their
- 557 corresponding URIs, concrete sub-profiles of this abstract profile could be defined for giving
- 558 support to their verification and update.
- When the requested form allows for different contents, the server MUST decide the specific
- 560 contents of the updated signature delivered, according to its configuration and settings.

561 3.5 Element < VerifyResponse >

This clause profiles the dss: VerifyResponse element.

563 3.5.1.1 Element < Optional Outputs>

- None of the optional inputs specified in the [DSS Core] are precluded in this abstract profile. It
- only constrains some of them.

566 3.5.1.1.1 Optional Output < Updated Signature >

- This element SHALL contain a dss:SignatureObject element that SHALL NOT contain a
- 568 dss:TimeStamp element as a child.

4 XML Advanced Electronic Signatures concrete Profile

569

570

571	4.1	Overv	riew
572 573 574 575	Electro	onic Signatures rel	profile supports operations within each phase of the lifecycle of XML Advanced ature based on [XMLSig] such as specified in [XAdES]. It will then provide all ated to XAdES signatures that are specified in the abstract profile defined in
576	For the	e generat	tion of XAdES signatures, the following operations apply:
577	0	SignRe	equest. This operation supports requests for:
578		0	Generating predefined advanced signature forms as defined in [XAdES].
579 580 581 582		0	Generating XML signatures incorporating specific signed/unsigned properties whose combination does not fit any predefined XAdES signature form. In such cases, the form MUST have been defined in a proprietary specification and MUST be identified by one URI.
583	0	SignRe	sponse. This operation supports delivery of:
584		0	Predefined advanced signature forms as defined in [XAdES].
585 586 587		0	XML signatures with specific properties whose combination does not fit any predefined XAdES signature form. In such cases, the form MUST have been defined in a proprietary specification and MUST be identified by one URI.
588	For ve	rification	[and updating] of XAdES signatures the following operations apply:
589	0	VerifyR	lequest. This operation supports requests for:
590		0	Verifying a predefined XAdES signature form.
591 592		0	Verifying XML signatures incorporating specific properties whose combination does not fit any predefined XAdES signature form.
593 594 595		0	Verifying any of the signatures mentioned above PLUS updating them by adding unsigned properties (time-stamps, validation data, etc) leading to a predefined XAdES form.
596		0	Verifying a long-term advanced signature in a certain point of time.
597	0	VerifyR	esponse. This operation supports delivery of:
598		0	Advanced signature verification result of signatures mentioned above.
599 600		0	Advanced signature verification result PLUS the updated signatures as requested.
601	4.2	Profile	e features
602	4.2.1	ldent	ifier
603	urn:	oasis:na	ames:tc:dss:1.0:profiles:XAdES.

oasis-dss-profiles-AdES-spec-cs-v1.0-r1

13 February 2007

604 **4.2.2 Scope**

This document profiles the DSS abstract profile defined in section 3 of the present document.

606 4.2.3 Relationship To Other Profiles

- The profile in this section is based on the abstract profile for Advanced Electronic Signatures
- 608 defined in section 3.

609 4.2.4 Signature Object

- This profile supports the creation and verification of XML advanced signatures as defined in
- 611 [XAdES].
- This profile also supports verification and update of advanced signatures by addition of
- 613 unsigned properties (time-stamps and different types of validation data), as specified in
- 614 [XAdES]

619

621

622 623

624

636

615 4.2.5 Transport Binding

This profile does not specify or constrain the transport binding.

617 **4.2.6 Security Binding**

This profile does not specify or constrain the security binding.

4.3 Profile of Signing Protocol

- 620 The present profile allows requesting:
 - Predefined forms of advanced electronic signatures as defined in [XAdES]. A server aligned with this profile SHALL generate XAdES signatures with direct incorporation of qualifying properties as defined in [XAdES] section 6.3.
 - Other forms of signatures based in [XMLSig] defined in other specifications,
- In both cases, the specific requested form will be identified by an URI.
- According to this profile, the following predefined advanced signature forms defined in
- 627 [XAdES] MAY be requested: XAdES-BES, XAdES-EPES, XAdES-T, XAdES-C, XAdES-X,
- 628 XAdES-X-L., and XAdES-A.
- In addition, the present profile provides means for requesting incorporation in any of the
- aforementioned forms any of the following properties: SigningTime,
- 631 CommitmentTypeIndication, SignatureProductionPlace, SignerRole,
- 632 IndividualDataObjectTimeStamp, AllDataObjectTimeStamp and
- 633 DataObjectFormat.
- 634 Other electronic signature forms based in [XMLSig] defined elsewhere MAY also be
- requested using the mechanisms defined in this profile.

4.3.1 Attribute Profile

037 urn:oasis:names:tc:dss:1.0:profiles:XAdES.

639	This clause profiles the dss:SignRequest element.
640	4.3.2.1 Element <optionalinputs></optionalinputs>
641	4.3.2.1.1 New Optional Inputs
642 643	4.3.2.1.1.1 Element <signatureform> Usage of these elements is according to what is stated in section 3.3.1.1.1.1.</signatureform>
644	4.3.2.1.2 Optional Inputs already defined in the Core
645 646	None of the optional inputs specified in the [DSS Core] are precluded in this abstract profile only constrains some of them and specifies additional optional inputs.
647	4.3.2.1.2.1 Optional Input <signaturetype></signaturetype>
648	This element is MANDATORY. Its vaule MUST be:
649	urn:ietf:rfc:3275
650	4.3.2.1.2.2 Optional inputs < ClaimedIdentity> and <keyselector></keyselector>
651	Usage of these elements is according to what is stated in section 3.3.1.1.2.2.
652	4.3.2.1.2.3 Optional Input <signedproperties></signedproperties>
653	4.3.2.1.2.3.1 Requesting SigningTime
654 655	Clients MAY use the URI defined in 3.3.1.1.2.3.1 or alternatively they MAY also use the following one:
656	urn:oasis:names:tc:dss:1.0:profiles:XAdES:SigningTime
657	Usage of these elements is according to what is stated in section 3.3.1.1.2.3.1.
658	4.3.2.1.2.3.2 Requesting CommitmentTypeIndication
659 660	Clients MAY use the URI defined in 3.3.1.1.2.3.2 or alternatively they MAY also use the following one:
661 662	urn:oasis:names:tc:dss:1.0:profiles:XAdES:CommitmentTypeIndication
663 664 665	When this optional input is present, the <value> element MUST contain a <requestedcommitment> element as defined in section in 3.3.1.1.2.3.2 with the <xades:commitmenttypeindication>.</xades:commitmenttypeindication></requestedcommitment></value>
666	4.3.2.1.2.3.3 Requesting SignatureProductionPlace
667 668	Clients MAY use the URI defined in 3.3.1.1.2.3.3 or alternatively they MAY also use the following one:
669 670	<pre>urn:oasis:names:tc:dss:1.0:profiles:XAdES:SignatureProductionPlac e</pre>

4.3.2 Element <SignRequest>

638

lt

- When this optional input is present, the <Value> element MUST contain a
- 672 <RequestedSignatureProductionPlace> element as defined in section 3.3.1.1.2.3.3
- 673 with the <xades:SignatureProductionPlace>.

674 4.3.2.1.2.3.4 Requesting SignerRole

- 675 Clients MAY use the URI defined in 3.3.1.1.2.3.4 or alternatively they MAY also use the
- 676 following one:

urn:oasis:names:tc:dss:1.0:profiles:XAdES:SignerRole

- When this optional input is present, the <Value> element MUST contain a
- 679 <RequestedSignerRole> element as defined in section 3.3.1.1.2.3.4 with the
- 680 < xades: SignerRole > child.

4.3.2.1.2.3.5 Requesting AllDataObjectTimeStamp

- 682 Clients MAY use the URI defined in 3.3.1.1.2.3.5 or alternatively they MAY also use the
- 683 following one:

684 urn:oasis:names:tc:dss:1.0:profiles:XAdES:AllDataObjectsTimeStamp

Usage of these elements is according to what is stated in section 3.3.1.1.2.3.5.

4.3.2.1.2.3.6 Requesting DataObjectFormat

- 687 Clients MAY use the URI defined in 3.3.1.1.2.3.6 or alternatively they MAY also use the
- 688 following one:

695

696

698

089 urn:oasis:names:tc:dss:1.0:profiles:XAdES:AllDataObjectsTimeStamp

- 690 When this optional input is present, the <Value> element MUST contain a
- 691 <RequestedDocsFormat> element as defined in section 3.3.1.1.2.3.6 with one or more
- 692 < DocFormat > children.

693 4.3.2.1.2.3.7 Requesting < xades: Individual Data Object Time Stamp >

694 Value for <Identifier> element:

urn:oasis:names:tc:dss:1.0:profiles:XAdES:IndividualDataObjectTimeStamp

In this case, the content of <Value> element will be the element

<DocsToBeTimeStamped>, defined as shown below.

```
<xs:element name="DocsToBeTimeStamped" type="DocReferencesType"/>
699
700
701
      <xs:complexType name="DocReferencesType">
702
         <xs:sequence>
703
            <xs:element name="DocReference" maxOccurs="unbounded"</pre>
704
               type="DocReferenceType"/>
705
         </xs:sequence>
706
      </xs:complexType>
707
708
      <xs:complexType name="DocReferenceType">
709
         <xs:attribute name="WhichDocument" type="xs:IDREF"</pre>
710
            use="required"/>
711
          <xs:attribute name="RefId" type="xs:string" use="optional"/>
712
      </xs:complexType>
```

- 713 WhichDocument attribute contains the reference to the document whose time-stamp is
- 714 requested (see attribute ID in [CoreDSS] section 2.4.1). Should the client request the
- 715 generation of several ds:Reference element for this document (using
- 716 dss:SignedReferences optional input), the server SHALL timestamp all the data objects
- 717 referenced by these ds:Reference elements. Under these conditions, each
- 718 dss:SignedReference element MUST have its RefId attribute set to a not empty value.
- 719 [XAdES] mandates that <ds:Reference> elements corresponding to signed data objects
- that have been individually time-stamped before being signed, must include an Id attribute.
- 721 [XAdES] also mandates < xades: IndividualDataObjectsTimeStamp > element to use
- 722 this Id attribute to indicate what signed documents have actually been time-stamped before
- 723 signing. See [XAdES] < xades: TimeStampType> and
- 725 The client MAY request a value for the <ds:Reference> element's Id attribute using the
- 726 Refid optional attribute if a <dss:SignedReference> forcing a value for such an attribute
- is not present in the request. If the request does not specify a value for this attribute, then the
- 728 server will automatically generate it.

729 4.3.3 Element <SignResponse>

730 This section profiles the dss:SignResponse element.

731 4.3.3.1 Element <SignatureObject>

- 732 The content of this element MUST be one of the following:
- 733 A ds:Signature element containing a XMLSig based signature.
- 734 A dss:SignaturePtr pointing to the XMLSig based signature embedded in an output
- 735 document.

736 4.4 Profile of Verifying Protocol

- 737 A server verifying XAdES signatures SHOULD follow the recommendations made by the
- 738 XAdES standard it aligns to with respect on how to verify the signed and unsigned properties
- 739 (version XAdES v1.3.2 includes an informative annex on this topic).

740 4.4.1 Element < VerifyRequest>

741 This clause profiles the dss: VerifyRequest element.

742 4.4.1.1 Attribute Profile

743 urn:oasis:names:tc:dss:1.0:profiles:XAdES.

744 4.4.1.2 Element <SignatureObject>

745 This element SHALL NOT contain a dss:TimeStamp element as a child.

746 4.4.1.3 Element < OptionalInputs>

747 4.4.1.3.1 Optional Output <ReturnUpdatedSignature>

Usage of these elements is according to what is stated in section 3.4.1.3.1.

749 4.4.2 Element < VerifyResponse>

750 This clause profiles the dss: VerifyResponse element.

751 4.4.2.1 Element < Optional Outputs>

- None of the optional inputs specified in the [DSS Core] are precluded in this profile. It only
- 753 constrains some of them.

754 4.4.2.1.1 Optional Output < Updated Signature >

- 755 The content of the dss:UpdatedSignature will be a dss:SignatureObject element
- with one of the following contents:
- o Ads:Signature containing a XMLSig based signature.
- 758 o A dss:SignaturePtr pointing to the XMLSig based signature embedded in one of the inputdocuments.

760 4.5 Profile Bindings

761 4.5.1 Transport Bindings

- 762 Messages transported in this profile MAY be transported by the HTTP POST Transport
- 763 Binding and the SOAP 1.2 Transport Binding defined in [DSSCore].

764 4.5.2 Security Bindings

765 4.5.2.1 Security Requirements

- 766 This profile MUST use security bindings that:
- o Authenticates the requester to the DSS server
- 768 o Authenticates the DSS server to the DSS client
- o Protects the integrity or a request, response and the association of response to the request.
- Optionally, protects the confidentiality of a request and response.
- 772 o The following MAY be used to meet these requirements.

773 4.5.2.2 TLS X.509 Mutual Authentication

- This profile is secured using the TLS X.509 Mutual Authentication Binding defined in
- 775 [DSSCore].

776

5 CMS-based Advanced Electronic Signature profile

777

808 809

810

778	p	rofile
779	5.1	Overview
780 781 782 783	Advano then pr	oncrete profile supports operations within each phase of the lifecycle of CMS based ched Electronic Signature based on [RFC 3852] such as specified in [CAdES]. It will ovide all the features related to CAdES signatures that are specified in the abstract defined in section 3.
784	For the	generation of CAdES signatures, the following operations apply:
785	0	SignRequest. This operation supports requests for:
786		 Generating predefined advanced signature forms as defined in [CAdES].
787 788 789 790		 Generating CMS signatures incorporating specific signed/unsigned attributes whose combination does not fit any predefined [CAdES] signature forms. In such cases, the form MUST have been defined in a proprietary specification and MUST be identified by one URI.
791	0	SignResponse. This operation supports delivery of:
792		 Predefined advanced signature forms as defined in [CAdES].
793 794 795 796		 CMS signatures incorporating specific signed attributes whose combination does not fit any predefined [CAdES] signature forms. In such cases, the form MUST have been defined in a proprietary specification and MUST be identified by one URI.
797 798	For ver	ification [and updating] of signatures as specified in [CAdES] the following operations
799	0	VerifyRequest. This operation supports requests for:
800		 Verifying a predefined [CAdES] signature form.
801 802		 Verifying CMS signatures incorporating specific attributes whose combination does not fit any predefined [CAdES] signature form.
803 804 805		 Verifying any of the signatures mentioned above PLUS updating them by addition of additional attributes (time-stamps, validation data, etc) leading to a predefined [CAdES] form.
806		 Verifying a long-term advanced signature in a certain point of time.
807	0	VerifyResponse. This operation supports delivery of:

Advanced signature verification result of signatures mentioned above.

Advanced signature verification result PLUS the updated signatures as

requested.

5.2 Profile features

812 **5.2.1 Identifier**

811

813 urn:oasis:names:tc:dss:1.0:profiles:CAdES.

814 **5.2.2 Scope**

This document profiles the DSS abstract profile defined in section 3 of the present document.

816 **5.2.3 Relationship To Other Profiles**

- 817 The profile in this document is based on the abstract profile for Advanced Electronic
- 818 Signatures defined in section 3.

819 5.2.4 Signature Object

- 820 This profile supports the creation and verification of CMS based advanced signatures as
- 821 defined in [CAdES].
- This profile also supports verification and update of advanced signatures by addition of
- 823 unsigned properties (time-stamps and different types of validation data), as specified in
- 824 [CAďES]

825 5.2.5 Transport Binding

This profile does not specify or constrain the transport binding.

827 **5.2.6 Security Binding**

This profile does not specify or constrain the security binding.

829 **5.3 Profile of Signing Protocol**

- 830 The present profile allows requesting:
- 831 o Predefined forms of advanced electronic signatures as defined in [CAdES].
- Other forms of signatures based in [RFC 3852] defined in other specifications,
- 833 In both cases, the specific requested form will be identified by an URI.
- 834 According to this profile, the following predefined advanced signature forms defined in
- 835 [CAdES] MAY be requested: CAdES-BES, CAdES-EPES, CAdES-T, CAdES-C, CAdES-X,
- 836 CAdES-X-L, and CAdES-A
- 837 In addition, the present profile provides means for requesting incorporation in any of the
- 838 aforementioned forms any of the following attributes: signing-time, commitment-type-
- 839 indication, signer-attributes, signer-location, content-hints, and
- 840 content-time-stamp
- Other electronic signature forms based in [RFC 3852], defined elsewhere, MAY also be
- requested using the mechanisms defined in this profile.

843 5.3.1 Element <SignRequest>

This clause profiles the dss:SignRequest element.

oasis-dss-profiles-AdES-spec-cs-v1.0-r1

13 February 2007

Copyright © OASIS Open 2007. All Rights Reserved.

845	5.3.1.1 Attribute Profile
846	urn:oasis:names:tc:dss:1.0:profiles:CAdES.
847	5.3.1.2 Element <optionalinputs></optionalinputs>
848	5.3.1.2.1 New Optional Inputs
849	5.3.1.2.1.1 Element <signatureform></signatureform>
850	Usage of these elements is according to what is stated in 3.3.1.1.1.
851	5.3.1.2.2 Optional Inputs already defined in the Core
852 853	None of the optional inputs specified in the [DSS Core] are precluded in this abstract profile. It only constrains some of them and specifies additional optional inputs.
854	5.3.1.2.2.1 Element <signaturetype></signaturetype>
855	This element is MANDATORY. Its value MUST be:
856	urn:ietf:rfc:3369
857	5.3.1.2.2.2 Optional inputs < ClaimedIdentity> / <keyselector></keyselector>
858	Usage of these elements is according to what is stated in section 3.3.1.1.2.2.
859	5.3.1.2.2.3 Element <signedproperties></signedproperties>
860	This section profiles section 3.3.1.1.2.3.
861	5.3.1.2.2.3.1 Requesting signing-time
862 863	Clients MAY use the URI defined in 3.3.1.1.2.3.1 or alternatively they MAY also use the following one:
864	urn:oasis:names:tc:dss:1.0:profiles:CAdES:signing-time
865	Usage of these elements is according to what is stated in section 3.3.1.1.2.3.1.
866	5.3.1.2.2.3.2 Requesting commitment-type-indication
867 868	Clients MAY use the URI defined in 3.3.1.1.2.3.2 or alternatively they MAY also use the following one:
869 870	urn:oasis:names:tc:dss:1.0:profiles:CAdES:commitment-type-indication
871 872 873 874	When this optional input is present, the <value> element MUST contain a <requestedcommitment> element as defined in section 3.3.1.1.2.3.2 with the <binaryvalue> child containing the base64encoding of commitment-type-indication ASN.1 attribute as specified in [CAdES], DER-encoded.</binaryvalue></requestedcommitment></value>
875	5.3.1.2.2.3.3 Requesting signer-location
876 877	Clients MAY use the URI defined in 3.3.1.1.2.3.3 or alternatively they MAY also use the following one:
878	urn:oasis:names:tc:dss:1.0:profiles:CAdES:signer-location
	oasis-dss-profiles-AdES-spec-cs-v1.0-r1 13 February 20

Copyright © OASIS Open 2007. All Rights Reserved.

Page 26 of 38

879	When this op	tional input is	present, the	<value></value>	element MUST	contain a
-----	--------------	-----------------	--------------	-----------------	--------------	-----------

- 880 < RequestedSignatureProductionPlace > element as defined in section 3.3.1.1.2.3.3
- 881 with the <BinaryValue> child containing the base64encoding of signer-location ASN.1
- attribute as specified in [CAdES], DER-encoded.

883 5.3.1.2.2.3.4 Requesting signer-attributes

- 884 Clients MAY use the URI defined in 3.3.1.1.2.3.4 or alternatively they MAY also use the
- 885 following one:

886 urn:oasis:names:tc:dss:1.0:profiles:CAdES:signer-attributes

- 887 When this optional input is present, the <Value> element MUST contain a
- 888 <RequestedSignerRole> element as defined in section 3.3.1.1.2.3.4 with the
- 889 <BinaryValue> child containing the base64encoding of signer-attributes ASN.1
- attribute as specified in [CAdES], DER-encoded.

891 **5.3.1.2.2.3.5 Requesting content-time-stamp**

- 892 Clients MAY use the URI defined in 3.3.1.1.2.3.5 or alternatively they MAY also use the
- 893 following one:

894 urn:oasis:names:tc:dss:1.0:profiles:CAdES:content-time-stamp

895 Usage of these elements is according to what is stated in section 3.3.1.1.2.3.5

896 5.3.1.2.2.3.6 Requesting content-hints

- 897 Clients MAY use the URI defined in 3.3.1.1.2.3.6 or alternatively they MAY also use the
- 898 following one:

urn:oasis:names:tc:dss:1.0:profiles:CAdES:content-hints

- 900 When this optional input is present, the <Value> element MUST contain a
- 901 <RequestedDocsFormat> element as defined in section 3.3.1.1.2.3.6 with the
- 902 <BinaryValue> child containing the base64 encoding of content-hints ASN.1 attribute
- 903 as specified in [CAdES], DER-encoded.

904 5.3.2 Element < SignResponse >

905 This section profiles the dss:SignResponse element.

906 5.3.2.1 Element <SignatureObject>

- 907 The dss:SignatureObject MUST contain the dss:Base64Signature child with a CMS
- 908 based signature base-64 encoded.

909 5.4 Profile of Verifying Protocol

910 5.4.1 Element < VerifyRequest>

911 This clause profiles the dss: VerifyRequest element.

912 **5.4.1.1 Attribute Profile**

913 urn:oasis:names:tc:dss:1.0:profiles:CAdES.

oasis-dss-profiles-AdES-spec-cs-v1.0-r1

914 5.4.1.2 Element < OptionalInputs>

915 **5.4.1.2.1 Element < Return Updated Signature >**

916 Usage of these elements is according to what is stated in section 3.4.1.3.1.

917 5.4.1.3 Element <SignatureObject>

- 918 The dss:SignatureObject element MUST contain the dss:Base64Signature child
- 919 with a CMS based signature base64 encoded.

920 5.4.2 Element < VerifyResponse>

921 This clause profiles the dss: VerifyResponse element.

922 5.4.2.1 Element < Optional Outputs>

Usage of these elements is according to what is stated in section 3.5.1.1.

924 5.4.2.1.1 Element < Updated Signature >

925 o The content of the dss:UpdatedSignature will be a dss:SignatureObject 926 element with a dss:Base64Signature element with the CMS based signature base64 927 encoded.

928 5.5 Profile Bindings

929 5.5.1 Transport Bindings

- 930 Messages transported in this profile MAY be transported by the HTTP POST Transport
- 931 Binding and the SOAP 1.2 Transport Binding defined in [DSSCore].

932 **5.5.2 Security Bindings**

933 5.5.2.1 Security Requirements

- 934 This profile MUST use security bindings that:
- 935 o Authenticates the requester to the DSS server
- 936 o Authenticates the DSS server to the DSS client
- 937 o Protects the integrity or a request, response and the association of response to the request.
- 939 o Optionally, protects the confidentiality of a request and response.
- 940 o The following MAY be used to meet these requirements.

941 5.5.2.2 TLS X.509 Mutual Authentication

This profile is secured using the TLS X.509 Mutual Authentication Binding defined in [DSSCore].

6 XML timestamps in XAdES signatures

- 945 XAdES specification [XAdES] defines a placeholder for incorporating XML timestamps within
- 946 XAdES signatures. As at the time [XAdES] was written no XML timestamps had been
- 947 specified, no details on their structure and management were included.
- 948 The current section provides rules for including XML timestamps into XAdES signatures. For
- 949 the rest of the present document a XML timestamp is a dss:Timestamp element as defined
- 950 in [DSSCore] section 5.1, incorporating a ds:Signature element profiled as indicated in
- 951 [DSSCore] section 5.1.1.

944

952

6.1 Generation and inclusion of XML timestamps

953 6.1.1 Profile for XAdES timestamp containers

- 954 [XAdES] defines the following timestamps containers:
- 955 xades:IndividualDataObjectTimeStamp, xades:AllDataObjectTimeStamp,
- 956 xades:SignatureTimeStamp, xades:RefsOnlyTimeStamp,
- 957 xades:SigAndRefsTimeStamp and xades:ArchiveTimeStamp.
- 958 XAdES timestamp containers MAY include more than one XML timestamp.
- 959 XAdES timestamp containers including XML timestamps will not use the explicit referencing
- 960 mechanism (the xades:Include element) defined in [XAdES] section 7.1.4.3.1.

961 962 963	The current document defines the structure of XML timestamps that timestamp more than one item in XAdES signatures i.e., all the timestamps defined in XAdES except the signature timestamp, which has already been profiled in [DSSCore] section 3.5.2.2.
964 965	6.1.2 XML timestamp within xades:IndividualDataObjectsTimeStamp
966 967	This timestamp will be included within xades: IndividualDataObjectsTimeStamp's xades: XMLTimeStamp child.
968	This timestamp must be compliant with the profile defined in [DSSCore] section 5.1.1.
969 970	In addition, this timestamp MUST include within its ds:SignedInfo one or more ds:Reference elements that will be built as indicate below.
971 972	1. Take all the XAdES signature's ds:Reference referencing those data objects designated by dss:DocsToBeTimestamped.
973	2. For each one proceed as indicated below:
974	a. Generate a copy.
975	b. Suppress the Id attribute of the copy if present.
976 977	c. Set the type attribute of the copy to the following URI: http://uri.etsi.org/01903/#IndividualDataObjectsTimeStamp.
978	d. Add the copy to the timestmp's ds: ds:SignedInfo.
979 980	Applications compliant with the present profile MUST dereference all the ds:Reference elements within XML timestamp's ds:SignedInfo as indicated in [XMLSig]
981	6.1.3 XML timestamp within xades:AllDataObjectsTimeStamp
982 983	This timestamp will be included within xades:AllDataObjectsTimeStamp'S xades:XMLTimeStamp child.
984	This timestamp must be compliant with the profile defined in [DSSCore] section 5.1.1.
985 986 987	In addition, this timestamp MUST include one ds:Reference element without URI attribute and the type attribute set to the following URI. http://uri.etsi.org/01903/#AllDataObjectsTimeStamp
988	It MUST NOT have any ds: Transforms element.
989 990 991 992	Applications compliant with the present profile MUST dereference this element by processing, as indicated in [XAdES] section 7.2.9 steps 1 to 3, all the ds:Reference elements in XAdES' ds:SignedInfo, except the one referencing the xades:SignedProperties element.
993	6.1.4 XML timestamp within xades:SigAndRefsTimeStamp
994 995	This timestamp will be included within xades:SigAndRefsTimeStamp's xades:XMLTimeStamp child.
996	This timestamp must be compliant with the profile defined in [DSSCore] section 5.1.1.
997 998 999	In addition, this timestamp MUST include one ds:Reference element without URI attribute and the type attribute set to the following URI. http://uri.etsi.org/01903/#SigAndRefsTimeStamp

oasis-dss-profiles-AdES-spec-cs-v1.0-r1

13 February 2007

- 1000 It MUST NOT have any ds: Transforms element.
- 1001 Applications compliant with the present profile MUST dereference this element by taking the
- data objects listed in [XAdES] section 7.5.1.1 and process them as indicated there.

1003 6.1.5 XML timestamp within xades:RefsOnlyTimeStamp

- 1004 This timestamp will be included within xades: RefsOnlyTimeStamp's
- 1005 xades:XMLTimeStamp child.
- 1006 This timestamp must be compliant with the profile defined in [DSSCore] section 5.1.1.
- 1007 In addition, this timestamp MUST include one ds:Reference element without URI attribute
- and the type attribute set to the following URI.
- 1009 http://uri.etsi.org/01903/#RefsOnlyTimeStamp
- 1010 It MUST NOT have any ds: Transforms element.
- 1011 Applications compliant with the present profile MUST dereference this element by the data
- 1012 objects listed in [XAdES] section 7.5.2.1 and process them as indicated there.

1013 **6.1.6 XML timestamp within xades:ArchiveTimeStamp**

- 1014 This timestamp will be included within xades: ArchiveTimeStamp's
- 1015 xades:XMLTimeStamp child.
- 1016 This timestamp must be compliant with the profile defined in [DSSCore] section 5.1.1.
- 1017 In addition, this timestamp MUST include one ds:Reference element without URI attribute
- and the type attribute set to the following URI.
- 1019 http://uri.etsi.org/01903/#ArchiveTimeStamp
- 1020 It MUST NOT have any ds: Transforms element.
- 1021 Applications compliant with the present profile MUST dereference this element by taking the
- data objects listed in [XAdES] section 7.7.1 and process them as indicated there.

1023 6.2 Verification of XML timestamps

- This section specifies the steps to be performed by a server for verifying the XML timestamps
- 1025 present in a XAdES signature.
- The steps that the server shall perform for initiating the verification of each XML timestamp
- 1027 within the corresponding container are listed in order below (if any one of them results in
- failure, then the timestamp token SHOULD be rejected).
- 1029 1. Extract the timestamp token embedded in the incoming signature.
- 1030 2. Verify that the verification key and algorithms used conforms to all relevant aspects of the applicable policy. Should this key come within a public certificate, verify that the certificate conforms to all relevant aspects of the applicable policy including algorithm usage, policy
- 1033 OIDs, and time accuracy tolerances.
- 1034 3. Verify that the aforementioned verification key is consistent with the 1035 ds:SignedInfo/SignatureMethod/@Algorithm attribute value.
- 1036 4. Verify the timestamp token signature in accordance with the rules defined in **[XMLDSIG]**.
- 1037 5. Verify that the ds:SignedInfo element contains only two ds:Reference elements

- 1038 6. Verify that one of the ds:Reference elements has its Type attribute set to 1039 "urn:oasis:names:tc:dss:1.0:core:schema:XMLTimeStampToken". Take this one and 1040 proceed as indicated below:
 - a. Retrieve the referenced data object. Verify that it references a ds:Object element, which in turn envelopes a dss:TSTInfo element.
 - b. Verify that the dss:TSTInfo element has a valid layout as per the present specification.
 - c. Extract the digest value and associated algorithm from its <ds:DigestValue> and <ds:DigestMethod> elements respectively.
 - d. Recalculate the digest of the retrieved data object as specified by **[XMLDSIG]** with the digest algorithm indicated in <ds:DigestMethod>, and compare this result with the contents of <ds:DigestValue>.
- Subsequent sub-sections indicate the steps that the server shall perform for completing the verification of each XML timestamp.

6.2.1 Verification of of xades:IndividualIDataObjectsTimeStamp including a XML timestamp

After completing steps 1 to 5 in section 6.2., the server will perform the tasks detailed below for completing the XML timestamp verification. If any one of them results in failure, then the timestamp token SHOULD be rejected. For each of the remaining ds:Reference proceed as indicated below:

- 1. Check that it has been built from one of the ds:Reference elements within XAdES signature applying the changes mentioned in section 6.1.2
- 1060 2. Dereference and validate it according to the rules stated in [XMLSig].
 - 3. Check for coherence in the value of the times indicated in the time-stamp tokens. All the time instants must be previous to the time when the verification is performed, to the time indicated within the SigningTime if present, and to the times indicated within the time-stamp tokens enclosed within all the rest of time-stamp container properties except other IndividualDataObjectsTimeStamp.
- 1066 4. Set the <dss:Result> element as appropriate.
- 1067 Minor Error

1041

1042

1043

1044 1045

1046 1047

1048

1049

1052

10531054

1055

1056 1057

1058

1059

1061 1062

1063

1064

1065

1071

10721073

1074 1075

1076

1077

1078

1079

1080

- 1068 urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidIndivid
 1069 ualDataObjectsTimestamp MUST be used when the cryptographic signature verification
- 1070 succeeds but this timestamp verification fails.

6.2.2 Verification of xades:AllDataObjectsTimeStamp including a XML timestamp

After completing steps 1 to 5 in section 6.2., the server will perform the steps listed below for completing the XML timestamp verification. If any one of them results in failure, then the timestamp token SHOULD be rejected.

- 1. Take the other ds:Reference element and proceed to dereference it as indicated below:
 - a. Take the first ds:Reference element within the XAdES signature's ds:SignedInfo element if and only if the Type attribute doesn"t have the value "http://uri.etsi.org/01903#SignedProperties".

- b. Process it according to the reference processing model of XMLDSIG.
- 1082 c. If the result is a node-set, canonicalize it using the algorithm indicated in CanonicalizationMethod element of the property, if present. If not, the standard canonicalization method as specified by XMLDSIG must be used.
 - d. Concatenate the resulting bytes in an octet stream.
 - e. Repeat steps a) to d) for all the subsequent ds:Reference elements (in their order of appearance) within the XAdES signature's ds:SignedInfo element if and only if Type attribute has not the value "http://uri.etsi.org/01903#SignedProperties".
 - f. Compute the digest of the resulting octet stream using the algorithm indicated in the time-stamp token and check if it is the same as the digest present there.
 - 2. Check for coherence in the value of the times indicated in the time-stamp tokens. All the time instants must be previous to the time when the verification is performed, to the time indicated within the SigningTime if present, and to the times indicated within the time-stamp tokens enclosed within all the rest of time-stamp container properties except IndividualDataObjectsTimeStamp.
- 1097 3. Set the <dss:Result> element as appropriate.
- 1098 Minor Error

1085

1086

1087 1088

1089 1090

1091 1092

1093

1094

1095

1096

1102 1103

1104

1105

11061107

1108

1111

1112 1113

1114

1115

1116

1117

1118

1119

1120 1121

- 1099 urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidAllData
- 1100 ObjectsTimestamp MUST be used when the cryptographic verification signature succeeds
- 1101 but this timestamp verification fails.

6.2.3 Verification of xades:SigAndRefsTimeStamp including a XML timestamp

After completing steps 1 to 5 in section 6.2, the server will perform the steps listed below for completing the XML timestamp verification. If any one of them results in failure, then the timestamp token SHOULD be rejected.

- 1. Check that those elements that, according to [XAdES] MUST be present for being timestamped by this timestamp, are actually present (see [XAdES] section 7.5.1).
- 1109 2. Take the other ds:Reference element and proceed to dereference it as indicated below:
 - a. Take the XAdES elements listed in [XAdES] section 7.5.1.1 in the order indicated there
 - b. Canonicalize them and concatenate the resulting bytes in one octet stream. If the CanonicalizationMethod element of the property is present, use it for canonicalizing. Otherwise, use the standard canonicalization method as specified by [XMLSiq].
 - c. Compute the digest of the resulting octet stream using the algorithm indicated in the time-stamp token and check if it is the same as the digest present there.
 - 3. Check that the time indicated by the timestamp is posterior to the one indicated in the xades:SigningTime property, and to the times indicated in the timestamps contained within xades:AllDataObjectsTimeStamp,
- 1122 xades:IndividualDataObjectsTimeStamp or xades:SignatureTimeStamp, if
 1123 present. They must also be previous to the times indicated in the timestamps enclosed by
 1124 any xades:ArchiveTimeStamp present elements
- 1125 Minor Error
- 1126 urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidSigAndR

1127 efsTimestamp MUST be used when the cryptographic verification signature succeeds but this timestamp verification fails.

1129 **6.2.4 Verification of xades:RefsOnlyTimeStamp including a XML** 1130 **timestamp**

After completing steps 1 to 5 in section 6.2, the server will perform the steps listed below for completing the XML timestamp verification. If any one of them results in failure, then the timestamp token SHOULD be rejected.

- 1. Check that those elements that, according to [XAdES] MUST be present for being timestamped by this timestamp, are actually present (see [XAdES] section 7.5.2).
- 1136 2. Take the other ds:Reference element and proceed to dereference it as indicated below:
 - a. Take the XAdES elements listed in [XAdES] section 7.5.2.1 in the order indicated there.
 - b. Canonicalize them and concatenate the resulting bytes in one octet stream. If the CanonicalizationMethod element of the property is present, use it for canonicalizing. Otherwise, use the standard canonicalization method as specified by [XMLSig].
 - c. Compute the digest of the resulting octet stream using the algorithm indicated in the time-stamp token and check if it is the same as the digest present there.
 - 3. Check that the time indicated by the timestamp is posterior to the one indicated in the xades:SigningTime property, and to the times indicated in the timestamps contained within xades:AllDataObjectsTimeStamp, xades:IndividualDataObjectsTimeStamp Or xades:SignatureTimeStamp, if present. They must also be previous to the times indicated in the timestamps enclosed by any xades:ArchiveTimeStamp present elements
- 1152 Minor Error

1134

1135

1138

1139

1140

1141 1142

1143

1144 1145

1146

1147

1148

1149

1150

1151

1156

11571158

1159

1160

1161

1162

1165

1166 1167

1168 1169

1170

1153 urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidRefsOnl 1154 yTimestamp MUST be used when the cryptographic verification signature succeeds but this 1155 timestamp verification fails.

6.2.5 Verification of xades:ArchiveTimeStamp including a XML timestamp

After completing steps 1 to 5 in section 6.2, the server will perform the steps listed below for completing the XML timestamp verification. If any one of them results in failure, then the timestamp token SHOULD be rejected.

- 1. Check that those elements that, according to [XAdES] MUST be present for being timestamped by this timestamp, are actually present (see [XAdES] section 7.7.1).
- 1163 2. Take the other ds:Reference element and proceed to dereference it as indicated below:
 - a. Take the XAdES elements listed in [XAdES] section 7.7.1 in the order indicated there.
 - b. Canonicalize them and concatenate the resulting bytes in one octet stream. If the CanonicalizationMethod element of the property is present, use it for canonicalizing. Otherwise, use the standard canonicalization method as specified by [XMLSig].

- 1171 c. Compute the digest of the resulting octet stream using the algorithm indicated in the time-stamp token and check if it is the same as the digest present there.
 - 3. Check that the time indicated by the timestamp is posterior to the one indicated in the SigningTime property, and to the times indicated in the timestamps contained within xades:AllDataObjectsTimeStamp, xades:IndividualDataObjectsTimeStamp, xades:SignatureTimeStamp if present, and xades:RefsOnlyTimeStamp or xades:SigAndRefsTimeStamp, if present They must also be previous to the times indicated in the timestamps enclosed by any xades:ArchiveTimeStamp that appear before the one that is being verified

1180 Minor Error

1173

1174 1175

1176

1177

1178

1179

1181 urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidArchive
1182 Timestamp MUST be used when the cryptographic verification signature succeeds but this
1183 timestamp verification fails.

7 Identifiers defined in this specification

7.1 Predefined advanced electronic signature forms identifiers

1187 The table below shows the URIs for standard forms of advanced electronic signature:

1188

1184

Advanced signature FORM	URI
XAdES-BES CAdES-BES	urn:oasis:names:tc:dss:1.0:profiles:AdES:forms:BES
XAdES-EPES CAdES-EPES	urn:oasis:names:tc:dss:1.0:profiles:AdES:forms:EPES
XAdES-T CAdES-ES-T	urn:oasis:names:tc:dss:1.0:profiles:AdES:forms:ES-T
XAdES-C CAdES-ES-C	urn:oasis:names:tc:dss:1.0:profiles:AdES:forms:ES-C
XAdES-X CAdES-ES-X	urn:oasis:names:tc:dss:1.0:profiles:AdES:forms:ES-X
XAdES-X-L CAdESX-L	urn:oasis:names:tc:dss:1.0:profiles:AdES:forms:ES-X-L
XAdES-A CAdES-X-A	urn:oasis:names:tc:dss:1.0:profiles:AdES:forms:ES-A

1189 Table 1.

1190

1191

7.2 Result Identifiers

This profile defines the <ResultMinor> values listed below. All of them indicate that the cryptographic verification of the signature succeeded, and that the verification of the indicated

timestamp failed.

1195 urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidIndivid

1196 ualDataObjectsTimestamp

1197 urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidAllData

1198 ObjectsTimestamp

oasis-dss-profiles-AdES-spec-cs-v1.0-r1

13 February 2007

Copyright © OASIS Open 2007. All Rights Reserved.

Page 36 of 38

1199 1200	urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidSigAndR efsTimestamp
1201 1202	urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidRefsOnlyTimestamp
1203 1204	urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidArchive Timestamp

A. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:
 Participants:

 Nick Pope, Thales eSecurity
 Ed Shallow, Universal Post Union
 Trevor Perrin, individual

1212

1205