



J2ME Code-Signing Profile of the OASIS Digital Signature Services

Committee Specification

13 February 2007

This Version:

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-codesigning-j2me-spec-cs-v0.1-r1.html>

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-codesigning-j2me-spec-cs-v0.1-r1.pdf>

Latest Version:

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-codesigning-j2me-spec-cs-v0.1-r1.html>

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-codesigning-j2me-spec-cs-v0.1-r1.pdf>

Technical Committee:

OASIS Digital Signature Services TC

Chair(s):

Nick Pope, Thales eSecurity

Juan Carlos Cruellas, Centre d'aplicacions avançades d'Internet (UPC)

<http://docs.oasis-open.org/dss/v1.0/>

Editor:

Andreas Kuehne, *individual*

Related work:

This specification is related to:

- [oasis-dss-core-spec-cs-v1.0-r1](#)

Abstract:

This document profiles the OASIS DSS core protocols and the OASIS DSS Abstract Code-Signing Profile for the purpose of creating J2ME code-signing signatures.

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.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical

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

45

Notices

46 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
47 that might be claimed to pertain to the implementation or use of the technology described in this
48 document or the extent to which any license under such rights might or might not be available;
49 neither does it represent that it has made any effort to identify any such rights. Information on
50 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
51 website. Copies of claims of rights made available for publication and any assurances of licenses
52 to be made available, or the result of an attempt made to obtain a general license or permission
53 for the use of such proprietary rights by implementors or users of this specification, can be
54 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 policies
59 apply.

60 This document and translations of it may be copied and furnished to others, and derivative works
61 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
62 published and distributed, in whole or in part, without restriction of any kind, provided that the
63 above copyright notice and this paragraph are included on all such copies and derivative works.
64 However, this document itself may not be modified in any way, such as by removing the copyright
65 notice or references to OASIS, except as needed for the purpose of developing OASIS
66 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
67 Property Rights document must be followed, or as required to translate it into languages other
68 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 OASIS
72 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
73 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
74 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
75 PARTICULAR PURPOSE.

76 The names "OASIS" are trademarks of OASIS, the owner and developer of this specification, and
77 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 enforce
79 its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for
80 above guidance.

81

Table of Contents

83	1	Introduction	5
84	1.1	Terminology.....	5
85	1.2	Namespaces	5
86	1.3	Normative References	5
87	1.4	Overview (Non-normative)	6
88	2	Profile Features.....	7
89	2.1	Identifier.....	7
90	2.2	Scope	7
91	2.3	Relationship To Other Profiles	7
92	2.4	Signature Object.....	7
93	2.5	Transport Binding.....	7
94	2.6	Security Binding	7
95	3	Profile of Signing Protocol.....	8
96	3.1	Element <dss:SignRequest>.....	8
97	3.1.1	Element <dss:OptionalInputs>.....	8
98	3.1.2	Element <dss:InputDocuments>.....	8
99	3.2	Element <dss:SignResponse>.....	9
100	3.2.1	Element <dss:Result>.....	9
101	3.2.2	Element <dss:OptionalOutputs>.....	9
102	3.2.3	Element <dss:SignatureObject>	10
103	4	Profile of Verifying Protocol.....	11
104	5	Profile of J2ME MIDP 2.0 Signatures	12
105	6	Profile of Server Processing Rules	13
106	7	Profile of Client Processing Rules	14
107		Appendix A. Acknowledgements	15

108

1 Introduction

109 The DSS signing and verifying protocols are defined in **[DSS Core]** and the code-signing profile
110 of the DSS signing and verification protocols are defined in **[DSS CS]**. As defined in those
111 documents, these protocols have a fair degree of flexibility and extensibility. This document
112 profiles these protocols to limit their flexibility and extend them in concrete ways. It also profiles
113 the processing rules followed by clients and servers when using these protocols, and profiles the
114 J2ME signature format for use with these protocols. The resulting profile is suitable for
115 implementation and interoperability.

116 The following sections describe how to understand the rest of this document.

1.1 Terminology

118 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
119 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be
120 interpreted as described in IETF RFC 2119 **[RFC 2119]**. These keywords are capitalized when
121 used to unambiguously specify requirements over protocol features and behavior that affect the
122 interoperability and security of implementations. When these words are not capitalized, they are
123 meant in their natural-language sense.

124 This specification uses the following typographical conventions in text: `<ns:Element>`,
125 Attribute, **Datatype**, OtherCode.

1.2 Namespaces

127 The structures described in this specification are contained in the schema file **[J2ME-CS-XSD]**.
128 All schema listings in the current document are excerpts from the schema file. In the case of a
129 disagreement between the schema file and this document, the schema file takes precedence.

130 This schema is associated with the following XML namespace:

```
131 urn:oasis:names:tc:dss:1.0:profiles:codesigning:1.0:J2ME:1.0
```

132 If a future version of this specification is needed, it will use a different namespace.

133 Conventional XML namespace prefixes are used in this document:

- 134 • The prefix `dssc:j2me:` (or no prefix) stands for the DSS code-signing namespace **[CS-**
135 **XSD]**.
- 136 • The prefix `dscs:` stands for the DSS code-signing namespace **[CS-XSD]**.
- 137 • The prefix `async:` stands for this profiles namespace **[Async-XSD]**.
- 138 • The prefix `dss:` stands for the DSS core namespace **[Core-XSD]**.
- 139 • The prefix `ds:` stands for the W3C XML Signature namespace **[XMLSig]**.

140 Applications MAY use different namespace prefixes, and MAY use whatever namespace
141 defaulting/scoping conventions they desire, as long as they are compliant with the Namespaces
142 in XML specification **[XML-ns]**.

1.3 Normative References

- | | | |
|-----|-------------------|---|
| 144 | [Core-XSD] | S Drees et al. <i>DSS Schema</i> . OASIS, February 2007 |
| 145 | [DSSCore] | S Drees et al. <i>Digital Signature Service Core Protocols and Elements</i> .
146 OASIS, February 2007 |

147	[RFC2119]	S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. IETF RFC 2119, March 1997.
148		
149		http://www.ietf.org/rfc/rfc2119.txt
150	[XML-ns]	T. Bray, D. Hollander, A. Layman. <i>Namespaces in XML</i> . W3C Recommendation, January 1999.
151		
152		http://www.w3.org/TR/1999/REC-xml-names-19990114
153	[XMLSig]	D. Eastlake et al. <i>XML-Signature Syntax and Processing</i> . W3C Recommendation, February 2002.
154		
155		http://www.w3.org/TR/1999/REC-xml-names-19990114
156	[DSS CS]	Abstract Code-Signing Profile of the OASIS Digital Signature Services Working Draft 03, 13 October 2004
157		
158	[DSS Async]	Asynchronous Processing Abstract Profile of the OASIS Digital Signature Services, Working Draft 04, 21 August 2004
159		
160	[CS-XSD]	P. Kasselmann, <i>Codesigning Schema</i> . OASIS, (MONTH/YEAR TBD)
161	[Async-XSD]	A. Kuehne. <i>Asynchronous Processing Profile Schema</i> . OASIS, (MONTH/YEAR TBD)
162		
163	[J2ME-CS-XSD]	P. Kasselmann, <i>J2ME Codesigning Schema</i> . OASIS, (MONTH/YEAR TBD)
164		
165	[MIDP 2.0]	Mobile Information Device Profile for Java™ 2 Micro Edition Version 2.0, JSR 118 Expert Group
166		
167	[RFC 2437]	RFC 2437 PKCS #1: RSA Cryptography Specifications Version 2.0, B. Kaliski, J. Staddon, http://www.ietf.org/rfc/rfc2437.txt
168		

169 **1.4 Overview (Non-normative)**

170 The **[DSS-CS]** abstract profile provides a profile of **[DSS-Core]** and combines it with the **[DSS-**
 171 **Async]** profile. The **[DSS-CS]** profile allow for the generation of signatures on content, including
 172 software programs, and is flexible enough to accommodate the typical scenarios encountered in
 173 the software development lifecycle.

174 This specification provides a concrete profile based on **[DSS-CS]** for requesting the generation of
 175 signatures as specified in the Java 2 Micro Edition (J2ME), Mobile Information Device Profile 2.0
 176 **[MIDP 2.0]**.

177 **2 Profile Features**

178 **2.1 Identifier**

179 **urn:oasis:names:tc:dss:1.0:profiles:codesigning:1.0:J2ME:1.0**

180 **2.2 Scope**

181 This document further profiles the abstract profile for code-signing as described in **[DSS CS]**,
182 which is a profile of the DSS signing protocol defined in **[DSS Core]** in combination with **[DSS**
183 **Async]**.

184 **2.3 Relationship To Other Profiles**

185 This profile is a concrete profile of the abstract code-signing profile defined in **[DSS CS]**.

186 **2.4 Signature Object**

187 This profile supports the creation of signatures as defined in **[MIDP 2.0]**. **[MIDP 2.0]** defines the
188 use of EMSA-PKCS1-v1_5 as defined in **[RFC 2437]**.

189 **2.5 Transport Binding**

190 This profile is transported using the HTTP POST Transport Binding defined in **[DSS Core]**.

191 **2.6 Security Binding**

192 This profile is secured using the TLS X.509 Mutual Authentication Binding defined in **[DSS Core]**.

193 3 Profile of Signing Protocol

194 3.1 Element <dss:SignRequest>

195 3.1.1 Element <dss:OptionalInputs>

196 Optional inputs MUST be used as defined in [DSS CS].

197 The following optional inputs defined in the [DSS Core] will not be understood by a server
198 implementing this profile:

- 199 • <dss:AddTimeStamp>
- 200 • <dss:SignedReference>
- 201 • <dss:Properties>
- 202 • <dss:SignaturePlacement>
- 203 • <dss:EnvelopingSignature>

204 In addition the following constraints are placed on the optional inputs as described below.

205 3.1.1.1 Element <dss:SignatureType>

206 The <dss:SignatureType> MUST contain the identifier `urn:ietf:rfc:2437:RSASSA-`
207 `PKCS1-v1_5`. This refers to PKCS #1 version 1.5 signatures as defined in [RFC 2437].

208 3.1.1.2 Element <dss:ServicePolicy>

209 The <dss:ServicePolicy> SHOULD be used to indicate a specific server signing policy. The
210 server signing policy is mapped to the recommended security policy for GSM/UMTS compliant
211 devices in [MIDP 2.0]. The following URIs may be used to specify the service policy and
212 corresponding domain under which the MIDlet must be signed.

213 For code that should execute in the manufacturer domain use:

214 `urn:oasis:names:tc:dss:1.0:profiles:codesigning:1.0:J2ME:1.0:manufactur`
215 `er`

216 For code that should execute in the operator domain use:

217 `urn:oasis:names:tc:dss:1.0:profiles:codesigning:1.0:J2ME:1.0:operator`

218 For code that should execute in the trusted third party domain use:

219 `urn:oasis:names:tc:dss:1.0:profiles:codesigning:1.0:J2ME:1.0:trustedisv`

220 3.1.2 Element <dss:InputDocuments>

221 The server MUST accept <dss:Document> inputs and MUST NOT accept
222 <dss:DocumentHash> inputs. A server that implements this profile MUST respond with a
223 <dss:ResultMajor> code of

224 `urn:oasis:names:tc:dss:1.0:resultmajor:RequesterError` as defined in [DSS
225 **Core**] if it receives a <dss:DocumentHash> input.

226 The <dss:Document> element MUST include the Base64 encoded J2ME JAR file on which the
227 signature must be calculated within a <dss:Base64Data> element. The `MimeType` attribute

228 MUST be set to `application/java-archive`. Only one `<Document>` element MUST be
229 submitted.

230 **3.2 Element `<dss:SignResponse>`**

231 **3.2.1 Element `<dss:Result>`**

232 This profile defines no additional `<dss:ResultMinor>` codes.

233 **3.2.2 Element `<dss:OptionalOutputs>`**

234 None of the optional outputs specified in the [DSS Core] are precluded in this abstract profile. In
235 addition this profile defines the following `<dss:OptionalOutputs>`:

- 236 • `<X509CertificatePath>`

237 In addition, the `<dss:OptionalOutputs>` element MAY contain a `<dss:Document>` element.

238 **3.2.2.1 Element `<X509CertificatePath>`**

239 This element defines the certificate path including the certificate containing the public key
240 required to verify the signature generated on the JAR file submitted by the client and all
241 intermediary certificates, excluding the root certificate. The client MAY use this information to
242 determine the appropriate entries in the Java Application Descriptor file (JAD) file that is
243 distributed with the JAR file containing the MIDP 2.0 application. The server may return multiple
244 `<X509CertificatePath>` elements. The orders of the `<X509CertificatePath>` elements are
245 significant. The first `<X509CertificatePath>` element corresponds to the first certificate path,
246 identified by $n=1$ in the JAD file, the second `<X509CertificatePath>` element corresponds to
247 the second certificate path, identified by $n=2$, in the JAD file, the j 'th `<X509CertificatePath>`
248 element corresponds to the j 'th certificate path, identified by $n=j$, in the JAD file. The
249 `<X509CertificatePath>` element contains the following elements:

250 `<X509Certificate>`

251 The `<X509Certificate>` element contains a base64-encoded X.509 v3 certificate.
252 The order of the `<X509Certificate>` elements are significant. The first
253 `<X509Certificate>` element contains the signing certificate and corresponds to $m=1$
254 in the JAD file for the current `<X509CertificatePath>` element, the second
255 `<X509Certificate>` element contains the first intermediary certificate and
256 corresponds to $m=2$ the current `<X509CertificatePath>` element, the k 'th
257 `<X509Certificate>` element contains the $k-1$ 'st intermediary certificate that issued
258 the $k-2$ 'nd intermediary cert.

259

```
260 <xs:element name="X509CertificatePath"  
261         type="dsscsj2me:X509CertificatePathType"/>  
262  
263 <xs:complexType name="X509CertificatePathType">  
264   <xs:sequence maxOccurs="unbounded">  
265     <xs:element ref="dsscsj2me:X509Certificate"/>  
266   </xs:sequence>  
267 </xs:complexType>
```

268

```
269 <xs:element name="X509Certificate"  
270         type="dsscsj2me:X509CertificateType"/>  
271  
272 <xs:simpleType name="X509CertificateType">  
273   <xs:restriction base="xs:base64Binary"/>  
274 </xs:simpleType>
```

275 **3.2.2.2 Element <dss:Documents>**

276 The server MAY include the J2ME JAR file on which the signature was created as an optional
277 output using the <dss:Documents> element. If the <dss:Document> element is included in
278 the response as an optional output, it MUST include the Base64 encoded J2ME JAR file within a
279 <dss:Base64Data> element. The included J2ME JAR file MUST be the file on which the
280 signature included in the <dss:SignatureObject> was calculated. The `MimeType` attribute
281 MUST be set to `application/java-archive`.

282 **3.2.3 Element <dss:SignatureObject>**

283 The server MUST return a Base64 encoded PKCS #1 signature within the <Base64Signature>
284 element. The <dss:SignatureObject> element MUST NOT contain any other elements.

285 **4 Profile of Verifying Protocol**

286 This **[DSS CS]** profile does not provide a profile of the DSS verification messages and
287 consequently a server implementing this profile **MUST NOT** respond to any
288 `<dss:VerifyRequest>` messages.

289 **5 Profile of J2ME MIDP 2.0 Signatures**

290 The J2ME MIDP 2.0 signature format is fully defined in **[MIDP 2.0]** and no further profiling is
291 required.

292

293

294

295

6 Profile of Server Processing Rules

296
297

The signature must be calculated on the Base64 decoded JAR file. The server processing rules defined in **[DSS CS]** SHOULD be followed.

298 **7 Profile of Client Processing Rules**
299 Client processing rules as defined in [DSS CS] SHOULD be followed.
300

301 **Appendix A. Acknowledgements**

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

304 **Participants:**

305 Trevor Perrin, *individual*

306 Pieter Kasselmann, Cybertrust

307