



Web Services Security: Non-Repudiation

Proposal Draft 05, 11 April 2003

Document identifier:

web-services-non-repudiation-05 ([PDF](#), [Word](#))

Location:

<http://schemas.reactivity.com/2003/04/web-services-non-repudiation-05.pdf>

Editor:

Eric L. Gravengaard, Reactivity, Inc. <eric@reactivity.com>

Contributors:

Grant Goodale, Reactivity
Michael Hanson, Reactivity
Brian Roddy, Reactivity
Dan Walkowski, Reactivity

Abstract:

This specification defines a method for requesting and sending message disposition receipts inside SOAP message headers. This specification makes use of the Web Services Security: SOAP Message Security and XML Digital Signature specifications.

Status:

This document is a proposal draft. Comments should be directed to the editor.

Copyright © 2003 Reactivity, Inc [Reactivity] All rights reserved

License:

Reactivity hereby grants you permission to copy and display the Web Services Security: Non-Repudiation [WSNR] Specification, in any medium without fee or royalty, provided that you include both the location and the copyright notice, both as printed above.

The furnishing of this specification does not grant you any rights or licenses, either expressly or by implication, in any intellectual property owned or controlled by Reactivity or any other party, whether necessary to implement the specification or otherwise. This document and the information contained herein is provided on an "AS IS" basis and REACTIVITY DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

The WSNR Specification may change without notice and you are cautioned against relying on the content of this specification.

The name and trademarks of the Authors may NOT be used in any manner, including advertising or publicity pertaining to the Specification or its contents without specific, written prior permission. Title to copyright in the WSNR specification will at all times remain with the authors.

40 **Table of Contents**

41	1	Introduction	3
42	2	Terminology	3
43	2.1	Namespaces	3
44	2.2	Glossary of Terms	4
45	3	Non-Normative Requirements	4
46	4	ReceiptRequest Element	4
47	4.1	ReceiptTo Element.....	6
48	5	Receipt Element.....	7
49	6	Use of Digital Signatures	8
50	6.1	SignatureRequest Element	8
51	6.2	SignatureResponse Element	9
52	6.3	Non-Normative Processing Model	10
53	7	Global Attributes	10
54	7.1	ReceiptFormat Attribute	10
55	7.2	CorrelationId Attribute	10
56	8	Error Handling	11
57	9	Security Considerations	11
58	10	Non-Normative Example.....	12
59	10.1	Simple Example	12
60	10.1.1	Request.....	12
61	10.1.2	Response.....	12
62	10.2	Signed Example	12
63	10.2.1	Request.....	12
64	10.2.2	Response.....	13
65	10.3	Full Example with Security Precautions.....	13
66	10.3.1	Request.....	13
67	10.3.2	Response.....	15
68	11	References.....	16
69	11.1	Normative	16
70	11.2	Non-Normative	17
71		Appendix A. Revision History	18
72		Appendix B. Notices	19
73			

74 1 Introduction

75 The Web Services Security: SOAP Message Security specification **[WSS]** defines the usage of
76 XML Digital Signatures within a SOAP header element to prove the integrity of a SOAP message.
77 While this is useful in the context of non-repudiation to the receiver, it does nothing to guarantee
78 to the sender that the message was delivered properly and without modification. Similarly, when
79 the SOAP requestor receives the SOAP response message there is no way of proving that the
80 SOAP response was generated after receiving and processing the SOAP request.

81 This specification extends the use of XML Digital Signature in the context of WSS: SOAP
82 Message Security to allow senders of SOAP messages to request message disposition
83 notifications that may optionally be signed to prove that the receiver received the SOAP message
84 without modification. The specification also defines a method for embedding SOAP message
85 dispositions in a SOAP message header. This specification constitutes a protocol for voluntary
86 non-repudiation of receipt that when used systematically provides cryptographic proof of both
87 parties participation in a transaction. This specification does not define any mechanism to prove
88 receipt of a message by a non-conformant implementation.

89 2 Terminology

90 The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT,
91 RECOMMENDED, NOT RECOMMENDED, MAY, and OPTIONAL in this document are to be
92 interpreted as described in **[RFC2119]**.

93 This specification is designed to work with the general SOAP message structure and message
94 processing model, and should be applicable to any version of SOAP. The current SOAP 1.2
95 namespace URI is used herein to provide detailed examples, but there is not intention to limit the
96 applicability of this specification to a single version of SOAP.

97 2.1 Namespaces

98 The following XML namespace URI MUST be used by implementations of this specification is as
99 follows:

100 `http://schemas.reactivity.com/2003/04/wsnr`

101 The following namespaces are used in this document:

Prefix	Namespace
s12	http://www.w3.org/2002/12/soap-envelope
ds	http://www.w3.org/2000/09/xmldsig#
wss	http://schemas.xmlsoap.org/ws/2003/03/secext
wsu	http://schemas.xmlsoap.org/ws/2002/07/utility
xs	http://www.w3.org/2001/XMLSchema

102 2.2 Glossary of Terms

103 Actor

104 An *actor* is any processor, the requestor, ultimate destination, or SOAP intermediary,
105 which receives and processes a SOAP message.

106 Integrity

107 *Integrity* is the property that data has not been modified.

108 Message Disposition Notification

109 *Message Disposition Notification* is a message reporting the status of a message. It
110 conveys information about whether the message was received and possibly if its integrity
111 was preserved in transit.

112 Signature

113 A *signature* is a value computed with a cryptographic algorithm and bound to data in such
114 a way that intended recipients of the data can use the signature to verify that the data has
115 not been altered since it was signed by the signer [XMLDSIG].

116 SOAP Intermediary

117 A *SOAP intermediary* is an application that is capable of both receiving and forwarding
118 SOAP messages.

119 SOAP Message Requestor

120 The *SOAP Message Requestor* is the originator of a SOAP Message and the client in the
121 HTTP Protocol binding defined in SOAP 1.1[SOAP11].

122 SOAP Message Responder

123 The *SOAP Message Responder* is the ultimate receiver of a SOAP Message and the
124 server in the HTTP Protocol binding defined in SOAP 1.1[SOAP11].

125 3 Non-Normative Requirements

126 This specification was designed to satisfy four requirements:

- 127 1. SOAP Message Requestors must be able to request a receipt for the SOAP Message
128 that is being transmitted.
- 129 2. SOAP Message Responders must be able to send a receipt for a SOAP Message that
130 requests one, either embedded in the SOAP Response or in another message.
- 131 3. SOAP Message Requestors must be able to specify what elements of the SOAP
132 Message they wish to have signed by the SOAP Message Responder.
- 133 4. SOAP Message receipts must be able to convey a signature for the elements that were
134 requested to be signed by the SOAP Message Requestor

135 4 ReceiptRequest Element

136 An actor uses the <ReceiptRequest> element to request that a subsequent actor send one or
137 more receipts for the message in which the <ReceiptRequest> is placed. The
138 <ReceiptRequest> element MUST be placed within a <wss:Security> element that has the
139 role attribute set such that the appropriate actor will process the <ReceiptRequest>. Each
140 <ReceiptRequest> element specifies the type of receipt and a set of one or more destinations

141 for the receipt to be sent. Multiple **<ReceiptRequest>** elements MAY be placed within the
142 same **<wss:Security>** element if multiple types of receipts are requested.

143 The syntax for this element is as follows:

```
144 <xs:element name="ReceiptRequest">  
145   <xs:complexType>  
146     <xs:sequence>  
147       <xs:element name="ReceiptTo" maxOccurs="unbounded">  
148         ...  
149       </xs:element>  
150       <xs:element ref="SignatureRequest" minOccurs="0"/>  
151       <xs:element ref="wsu:Timestamp" minOccurs="0"/>  
152     </xs:sequence>  
153     <xs:attribute ref="ReceiptFormat" use="required"/>  
154     <xs:attribute ref="CorrelationId" use="optional"/>  
155     <xs:attribute ref="wsu:Id" use="optional"/>  
156     <xs:attribute ref="S12:mustUnderstand" use="optional"/>  
157     <xs:anyAttribute/>  
158   </xs:complexType>  
159 </xs:element>
```

160 **/ReceiptRequest/@ReceiptFormat**

161 The **ReceiptFormat** attribute designates the type of receipt that is requested. The attribute is of
162 type **xs:anyURI**. The attribute is required. The legal values of **ReceiptFormat** are described in
163 section 7.1. When using either the **generalReceipt** or the **signedReceipt** formats, the
164 **<ReceiptRequest>** MAY contain a **<wsu:TimeStamp>** element. When using the
165 **signedReceipt** format, the **<ReceiptRequest>** MUST contain a **<SignatureRequest>**
166 element. Conformant implementations MUST be able to process both formats.

167 **/ReceiptRequest/@CorrelationId**

168 The optional **CorrelationId** attribute is used in the **<ReceiptRequest>** to specify a unique
169 identifier for the **<ReceiptRequest>**. When a **CorrelationId** is included in the request,
170 actors responding with receipts MUST include the **CorrelationId** of the request to allow the
171 requestor to match the request to the receipt. The type and recommended usage of
172 **CorrelationId** is specified in section 7.2.

173 **/ReceiptRequest/@wsu:Id**

174 The optional **wsu:Id** attribute is used to specify a unique identifier for the **<ReceiptRequest>**
175 element that can be used to reference the element. It is RECOMMENDED that this attribute be
176 used to allow the **<ReceiptRequest>** to be signed using an XML Digital Signature.

177 **/ReceiptRequest/@S12:mustUnderstand**

178 The optional **s12:mustUnderstand** attribute allows requestors to specify that a SOAP Fault
179 MUST be returned if the actor cannot process the **<ReceiptRequest>** element. The
180 **mustUnderstand** attribute MUST be in the same namespace of the root **<Envelope>** element.

181 **/ReceiptRequest/@any**

182 The **any** attribute is included in the schema for the purpose of satisfying the multiple namespaces
183 in which the **mustUnderstand** attribute may be described. This is the ONLY permitted use of
184 this schema extension.

185 **/ReceiptRequest/ReceiptTo**

186 One or more **<ReceiptTo>** elements are required to indicate where the receipt should be sent.
187 When multiple **<ReceiptTo>** elements are present, a copy of the **<Receipt>** MUST be sent to
188 each of the targets specified unless the **Required** element is set to false. The syntax for the
189 **<ReceiptTo>** element is described in section 4.1.

190 **/ReceiptRequest/SignatureRequest**

191 The <SignatureRequest> element MUST be present if and only if the ReceiptFormat
192 attribute equals signedReceipt. The syntax of this element is described in section 6.1.

193 **/ReceiptRequest/wsu:TimeStamp**

194 The optional <wsu:TimeStamp> element MAY be included to indicate the creation time of the
195 <ReceiptRequest> element. If the element is included, conformant implementations MUST
196 return a SOAP Fault if the creation time is in the future or if the expiration time is in the past.

197 **4.1 ReceiptTo Element**

198 The <ReceiptTo> element is used to convey information on how a responding actor should
199 send a receipt to the receipt requestor. The syntax for this element is as follows:

```
200 <xs:element name="ReceiptTo" maxOccurs="unbounded">
201   <xs:complexType>
202     <xs:attribute name="Target" type="xs:anyURI" use="optional"
203     default="http://schemas.reactivity.com/wsnr/2003/04/response"/>
204     <xs:attribute name="Required" type="xs:boolean" use="optional"
205     default="1"/>
206     <xs:attribute name="ReceiptAddress" type="xs:anyURI" use="optional"/>
207     <xs:attribute ref="S12:role" use="optional"
208     default="http://www.w3.org/2002/12/soap-envelope/role/ultimateReceiver"/>
209     <xs:anyAttribute/>
210   </xs:complexType>
211 </xs:element>
```

212 **/ReceiptRequest/ReceiptTo/@Target**

213 The optional Target attribute specifies where the <Receipt> should be sent. If this attribute is
214 omitted, then the default value, response, should be assumed. There can be only one
215 <ReceiptTo> element that omits the Target attribute or explicitly has Target equal to
216 response. The response option is only available for the request SOAP message in a two-
217 message request-response context. Clearly, a <ReceiptTo> element within an HTTP response
218 MUST NOT have Target equal to response and MUST NOT omit the Target attribute. Legal
219 values for Target are shown below:

Short name	Long name / Description
response (default)	http://schemas.reactivity.com/2003/04/wsnr/response
	The <Receipt> should be included in the <wss:Security> header of the response SOAP message.
HTTPS	http://schemas.reactivity.com/2003/04/wsnr/HTTPS
	A SOAP message with a <Receipt> in the <wss:Security> header should be sent to the URL indicated in the ReceiptAddress attribute.
SMTP	http://schemas.reactivity.com/2003/04/wsnr/SMTP
	A SOAP message with a <Receipt> in the <wss:Security> header should be sent to the email address indicated in the ReceiptAddress attribute.

220 If the value of Target is not response, then a ReceiptAddress attribute MUST be included in
221 the <ReceiptTo> element.

222 **/ReceiptRequest/ReceiptTo/@Required**

223 The optional Required attribute is used to indicate if the <Receipt> is required by the
224 requestor or if is optional. If the Required attribute is set to true, then the responding actor MUST
225 send either a <Receipt> or a SOAP Fault. If the Required attribute is set to false, then the

226 responding actor MAY return a `<Receipt>` depending on its own security policies. The default
227 value is true if the attribute is omitted.

228 **/ReceiptRequest/ReceiptTo/@ReceiptAddress**

229 The `ReceiptAddress` attribute MUST be specified if the `Target` attribute equals either
230 `HTTPSOAP` or `SMTPSOAP`. If the value of `Target` is `HTTPSOAP`, then the value of the
231 `ReceiptAddress` attribute MUST be a HTTP/S URL where a SOAP message containing a
232 `<wss:Security>` element containing a `<Receipt>` MUST be sent using the HTTP POST
233 protocol. If the value of `Target` is `SMTPSOAP`, then the value of the `ReceiptAddress` attribute
234 MUST be a `mailto:` URL that specifies an SMTP address where a SOAP message containing a
235 `<wss:Security>` element containing a `<Receipt>` MUST be sent.

236 **/ReceiptRequest/ReceiptTo/@S12:role**

237 The optional `S12:role` attribute specifies the value of role in the `<wss:Security>` header that
238 MUST contain the `<Receipt>`. Compliant implementations MUST either use an existing
239 `<wss:Security>` header with the corresponding `S12:role` attribute or insert a new
240 `<wss:Security>` header with the appropriate `S12:role` attribute. The `S12:role` attribute
241 SHOULD be specified in the same namespace as the corresponding outer `<Envelope>` to
242 enable the receiver to properly interpret this value and respond appropriately. The default value if
243 the attribute is not specified is `S12:ultimateReceiver`.

244 **/ReceiptRequest/ReceiptTo/@any**

245 The `any` attribute is included in the schema for the purpose of satisfying future namespaces in
246 which the `role` attribute may be described. This is the ONLY permitted use of this schema
247 extension.

248 5 Receipt Element

249 The `<Receipt>` element is used to respond to a `<ReceiptRequest>`. It conveys the fact that
250 the message to which the `<ReceiptRequest>` was attached to was received but not that it was
251 interpreted correctly or processed. Therefore, the `<Receipt>` may be issued even when there
252 was a problem processing the message that contained the `<ReceiptRequest>` so long as the
253 `<ReceiptRequest>` was processed correctly. For example, a `<Receipt>` element could be
254 returned in the event of a SOAP Fault. `<Receipt>` elements MUST be placed within a
255 `<wss:Security>` header element of a SOAP message. The syntax for the `<Receipt>` element
256 is as follows:

```
257 <xs:element name="Receipt">  
258   <xs:complexType>  
259     <xs:sequence>  
260       <xs:element ref="SignatureResponse" minOccurs="0"/>  
261       <xs:element ref="wsu:Timestamp" minOccurs="0"/>  
262     </xs:sequence>  
263     <xs:attribute ref="ReceiptFormat" use="required"/>  
264     <xs:attribute ref="CorrelationId" use="optional"/>  
265     <xs:attribute ref="wsu:Id" use="optional"/>  
266   </xs:complexType>  
267 </xs:element>
```

268 **/Receipt/@ReceiptFormat**

269 The `ReceiptFormat` attribute designates what type of `<Receipt>` is being sent. The attribute
270 is of type `xs:anyURI`. The attribute is required. The legal values of `ReceiptFormat` are
271 described in section 7.1. When using either the `generalReceipt` or the `signedReceipt`
272 formats, the `<Receipt>` MAY contain a `<wsu:TimeStamp>` element. When using the
273 `signedReceipt` format, the `<Receipt>` MUST contain a `<SignatureResponse>` element.

274 /Receipt/@CorrelationId

275 The optional `CorrelationId` attribute is used in the `<Receipt>` to specify the unique identifier
276 used to identify the `<ReceiptRequest>`. When a `CorrelationId` is included in the request,
277 actors responding with receipts **MUST** include the `CorrelationId` of the request to allow the
278 requestor to match the request to the receipt. The type and recommended usage of
279 `CorrelationId` is specified in section 7.2.

280 /Receipt/@wsu:Id

281 The optional `wsu:Id` attribute is used to specify a unique identifier for the `<Receipt>` element
282 that can be used to reference the element. It is **RECOMMENDED** that this attribute be used to
283 allow the `<Receipt>` to be signed using an XML Digital Signature.

284 /Receipt/SignatureResponse

285 The `<SignatureResponse>` element **MUST** be present if and only if the `ReceiptFormat`
286 attribute equals `signedReceipt`. The syntax of this element is described in section 6.2.

287 /Receipt/wsu:TimeStamp

288 The optional `<wsu:TimeStamp>` element **MAY** be included to indicate the creation time of the
289 `<Receipt>` element. The use of `<wsu:Expires>` element has no meaning in this context.

290 6 Use of Digital Signatures

291 Using digital signatures adds further capabilities to the use of `<Receipts>`. XML digital
292 signatures allow the receiving party to verify the integrity of the signed data and the identity of the
293 signing party. The format and processing instructions for XML digital signatures have been
294 defined by the W3C **[XMLDSIG]**. However a single signature generated by a single actor cannot
295

296 signature has been divided into two halves: that which should be signed, and the actual signature
297 value. The XML Digital Signature specification defines several components that can be used to
298 describe both of these components in greater detail. This specification defines two new elements,
299 `<SignatureRequest>` and `<SignatureResponse>`, to hold the sub-elements of a
300 `<ds:Signature>` that represent a request for signature and a response.

301 When combined, the sub-elements of a `<SignatureRequest>` and `<SignatureResponse>`,
302 form all the subcomponents of a `<ds:Signature>`.

303 6.1 SignatureRequest Element

304 The `<SignatureRequest>` element is used to request a digital signature that covers a dataset
305 specified by the `<ds:SignedInfo>` sub-element. The syntax for this element is as follows:

```
306 <xs:element name="SignatureRequest">  
307   <xs:complexType>  
308     <xs:sequence>  
309       <xs:element ref="ds:SignedInfo"/>  
310       <xs:element ref="ds:Object" minOccurs="0" maxOccurs="unbounded"/>  
311     </xs:sequence>  
312     <xs:attribute ref="wsu:Id" use="optional"/>  
313     <xs:anyAttribute/>  
314   </xs:complexType>  
315 </xs:element>
```

316 /SignatureRequest/@wsu:Id

317 The optional `wsu:Id` attribute is used to specify a unique identifier for the
318 `<SignatureRequest>` element that can be used to reference the element. It is intended to

319 allow the element to be signed using an XML Digital Signature if the element is not used in the
320 context of a `<ReceiptRequest>`.

321 `/SignatureRequest/@any`

322 The `any` attribute is used to allow this element to be used in contexts other than
323 `<ReceiptRequest>`. When used in the context of a `<ReceiptRequest>` element, no other
324 attributes are defined for the `<SignatureRequest>` element.

325 `/SignatureRequest/ds:SignedInfo`

326 The `<ds:SignedInfo>` element is used to convey information about what dataset the requestor
327 would like signed. The `<ds:SignedInfo>` element will also convey information about the state
328 of the dataset at the time of request because it will contain one or more `<ds:Reference>`
329 elements with a corresponding `<ds:DigestValue>`. The recipient of the
330 `<SignatureRequest>` SHOULD verify that the `<ds:SignedInfo>` is still valid and then
331 compute a `<ds:SignatureValue>`. The element is a required sub-element of
332 `<SignatureRequest>`.

333 `/SignatureRequest/ds:Object`

334 Zero or more `<ds:Object>` elements can be used to hold additional data for the signature to be
335 computed over. It is allowed here only to allow for the reuse of the `<SignatureRequest>`
336 element in other contexts. Since no processing will occur on `<ds:Object>` elements, it is NOT
337 RECOMMENDED that they be used in the context of a `<ReceiptRequest>`.

338 6.2 SignatureResponse Element

339 The `<SignatureResponse>` element is used to respond to a `<SignatureRequest>`. It
340 contains both the `<ds:SignatureValue>` and optional `<ds:KeyInfo>` elements that together
341 with the sub-elements of the original `<SignatureRequest>` comprise a complete
342 `<ds:Signature>`. The syntax for the `<SignatureResponse>` element is as follows:

```
343 <xs:element name="SignatureResponse">  
344   <xs:complexType>  
345     <xs:sequence>  
346       <xs:element ref="ds:SignatureValue"/>  
347       <xs:element ref="ds:KeyInfo" minOccurs="0"/>  
348     </xs:sequence>  
349     <xs:attribute ref="wsu:Id" use="optional"/>  
350     <xs:anyAttribute/>  
351   </xs:complexType>  
352 </xs:element>
```

353 `/SignatureResponse/@wsu:Id`

354 The optional `wsu:Id` attribute is used to specify a unique identifier for the
355 `<SignatureResponse>` element that can be used to reference the element. It is intended to
356 allow the element to be signed using an XML Digital Signature if the element is not used in the
357 context of a `<Receipt>`.

358 `/SignatureResponse/@any`

359 The `any` attribute is used to allow this element to be used in contexts other than `<Receipt>`.
360 When used in the context of a `<Receipt>`, no other attributes are defined for the
361 `<SignatureResponse>` element.

362 `/SignatureResponse/ds:SignatureValue`

363 The required `<ds:SignatureValue>` element conveys the value of the cryptographic signature
364 that covers the `<ds:SignedInfo>` element from the `<SignatureRequest>`.

365 **/SignatureResponse/ds:KeyInfo**
 366 The optional `<ds:KeyInfo>` element conveys information about the key used to compute the
 367 `<ds:SignatureValue>`. The element can contain any legal values as specified in the XML
 368 Digital Signature and Web Services Security specifications. It is RECOMMENDED that the
 369 `<ds:KeyInfo>` element contain a `<wss:SecurityTokenReference>` and that the key be
 370 prepended to the enveloping `<wss:Security>` header before the `<Receipt>` element.

371 **6.3 Non-Normative Processing Model**

372 To generate a `<SignatureRequest>`, the SOAP Message Requestor should first generate the
 373 SOAP envelope including the data contained in the `<S12:Body>`. Then the Requestor should
 374 create a `<ds:SignedInfo>`, including all references, for the data that the Responder should
 375 validate and for which it should respond with a disposition notification. Then the requestor should
 376 prepend the `<ReceiptRequest>` with the `<SignatureRequest>` into the applicable
 377 `<wss:Security>` header. A signature may then be created that references the
 378 `<ReceiptRequest>` and prepended to the `<wss:Security>` header.

379 When processing a `<ReceiptRequest>`, the Responder should first validate that the
 380 `<ds:SignedInfo>` is valid and that all of the `<ds:DigestValue>` elements are still valid. If
 381 they are not, then the Responder cannot attest to having received the data for which the
 382 Requestor asked to receive a disposition notification and therefore a proper receipt cannot be
 383 generated. If the digests are valid, then the Responder should calculate a
 384 `<ds:SignatureValue>`. This can then be inserted into a `<Receipt>` in the appropriate
 385 `<wss:Security>` header of the SOAP response message. A signature may then be created
 386 that references the `<Receipt>` and is prepended to the `<wss:Security>` header.

387 **7 Global Attributes**

388 The following attributes are used by multiple elements defined in this specification:

389 **7.1 ReceiptFormat Attribute**

390 The `ReceiptFormat` attribute designates the format of the `<Receipt>` or `<ReceiptRequest>`
 391 element. The attribute is of type `xs:anyURI`. The attribute is required for both `<Receipt>` and
 392 `<ReceiptRequest>` elements. There are two legal values for this attribute:

Short name	Long name	Description
<code>generalReceipt</code>	<code>http://schemas.reactivity.com/2003/04/wsnr/generalReceipt</code>	This format is for general unsigned receipts.
<code>signedReceipt</code>	<code>http://schemas.reactivity.com/2003/04/wsnr/signedReceipt</code>	This format is for signed receipts.

393 **7.2 CorrelationId Attribute**

394 The `CorrelationId` attribute is an `xs:string` that can be used to uniquely identify a pair of
 395 `<ReceiptRequest>` and `<Receipt>` elements. It MUST be unique to both the sender and the
 396 recipient so that each may log it and reference it later by this value. It is RECOMMENDED that
 397 the `CorrelationId` value be formatted as an `urn:uuid [UUID]`. For example:

398
399
400
401
402

```
<ReceiptRequest
  CorrelationId="urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
  ReceiptFormat="generalReceipt">
  <ReceiptTo/>
</ReceiptRequest>
```

403

8 Error Handling

404 If the Responder does not understand how to process a `<ReceiptRequest>` then the
405 Responder MUST return a SOAP Fault or stop processing. The **faultcode** for this class of error is
406 **S12:MustUnderstand**.

407 When using the `signedRequest` format, if the Responder cannot verify that the
408 `<ds:SignedInfo>` references are valid, then the Responder MUST NOT send a receipt. The
409 Responder MUST return a SOAP Fault or stop processing. The **faultcode** for this class of error is
410 **wsnr:InvalidSignedInfo**.

411

9 Security Considerations

412 There are three main security considerations when using this specification for secure non-
413 repudiation. First, both the receipt requestor and the receipt generator should keep secure
414 records of all message traffic. This is important because the complete signature is distributed
415 across both the request and the receipt and can only be verified when both pieces are present.
416 Only when both sides of an exchange log both pieces can both parties make any guarantee of
417 message disposition. Schneier and Kelsey present a cryptographic method for secure logging in
418 their 1999 paper [**Schneier**].

419 Second, both the `<ReceiptRequest>` and the `<Receipt>` elements should be signed. This
420 allows the receiving party to know that neither the `<ReceiptRequest>` nor the `<Receipt>`
421 were tampered with en route. In the case of the `<ReceiptRequest>`, this guarantees that the
422 `<ds:SignedInfo>` element was not changed to remove a key element from the dataset used
423 for the computation of the signature value.

424 Third, the trust relationship between two parties impacts the level of acceptance each party
425 should have for the other party's notion of time. As previously recommended, the
426 `<ReceiptRequest>` and `<Receipt>` elements should include a `<wsu:Timestamp>` element
427 indicating the time the encapsulating element was generated. If the encapsulating element is
428 digitally signed following the method described in Section 6, this timestamp may be taken at face
429 value in communications between parties with a medium to high degree of trust.

430 In communications between parties with a low degree of trust, a trusted digital time stamping
431 service capable of producing digitally signed timestamps in a format understood by both parties
432 should be used. The signed timestamp should at a minimum contain the digest of the
433 `<ReceiptRequest>` element and all elements referenced within the receipt request. In any
434 event, timestamps containing future times or times that differ from the receiving party's notion of
435 the current time should be treated as highly suspect.

436 10 Non-Normative Example

437 10.1 Simple Example

438 10.1.1 Request

```
439 <wsse:Security>
440   <ReceiptRequest ReceiptFormat="generalReceipt" CorrelationId="33485">
441     <ReceiptTo Required="true" Target="response"/>
442     <wsu:Timestamp>
443       <wsu:Created>2003-03-11T16:30:17Z</wsu:Created>
444     </wsu:Timestamp>
445   </ReceiptRequest>
446 </wsse:Security>
```

447 10.1.2 Response

```
448 <wsse:Security>
449   <Receipt ReceiptFormat="generalReceipt" CorrelationId="33485">
450     <wsu:Timestamp>
451       <wsu:Received>2003-03-11T16:33:43Z</wsu:Received>
452     </wsu:Timestamp>
453   </Receipt>
454 </wsse:Security>
```

455 10.2 Signed Example

456 10.2.1 Request

```
457 <S:Envelope xmlns:S="...">
458   <S:Header>
459     <wsse:Security>
460       <wsnr:ReceiptRequest ReceiptFormat="signedReceipt"
461         Role="ultimateReceiver" CorrelationID="theID"
462         S:mustUnderstand="1">
463         <wsnr:ReceiptTo Target="response">
464           <wsnr:SignatureRequest>
465             <ds:SignedInfo>
466               <ds:CanonicalizationMethod Algorithm="#c14n"/>
467               <ds:SignatureMethod Algorithm="#hmac-sha1"/>
468               <ds:Reference URI="#body">
469                 <ds:DigestMethod Algorithm="#sha1"/>
470               </ds:Reference>
471               <ds:Reference URI="#timestamp">
472                 <ds:DigestMethod Algorithm="#sha1"/>
473               </ds:Reference>
474             </ds:SignedInfo>
475           </wsnr:SignatureRequest>
476         </wsnr:ReceiptTo>
477         <wsu:Timestamp wsu:Id="timestamp">
478           <wsu:Created>2003-03-11T08:42:00Z</wsu:Created>
479         </wsu:Timestamp>
480       </wsnr:ReceiptRequest>
481     </wsse:Security>
482   </S:Header>
483   <S:Body>
484     <MyRequest wsu:Id="body"/>
485   </S:Body>
486 </S:Envelope>
```

487

10.2.2 Response

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

```

<S:Envelope xmlns:S="...">
  <S:Header>
    <wsse:Security S:Role="ultimateReceiver">
      <wsse:BinarySecurityToken wsu:Id="#theCert"
        EncodingType="Base64Binary">
        MIEZzCCA9CgAWIQEmtJZco...
      </wsse:BinarySecurityToken>
      <wsnr:Receipt ReceiptFormat="signedReceipt"
        CorrelationID="theID">
        <wsnr:SignatureResponse>
          <ds:SignatureValue>
            ABCDEFG1234567890...
          </ds:SignatureValue>
          <ds:KeyInfo>
            <wsse:SecurityTokenReference>
              <wsse:Reference URI="#theCert"/>
            </wsse:SecurityTokenReference>
          </ds:KeyInfo>
        </wsnr:SignatureResponse>
        <wsu:Timestamp>
          <wsu:Received>2003-03-11T08:42:12Z</wsu:Received>
        </wsu:Timestamp>
      </wsnr:Receipt>
    </wsse:Security>
  </S:Header>
  <S:Body>
    <MyResponse/>
  </S:Body>
</S:Envelope>

```

517

10.3 Full Example with Security Precautions

518

519

The following example shows the non-normative recommended usage of this specification to securely request and send a receipt using the `signedReceipt` format.

520

10.3.1 Request

521

522

523

524

525

526

527

The SOAP Message Requestor generates a `<ds:SignedInfo>` element that references and digests the `<S12:Body>` and the `<wsu:Timestamp>` elements. The `<ReceiptRequest>` element is then signed. The Requestor does not need to additionally sign the `<S12:Body>` element because that is covered by the signed `<ds:SignedInfo>` of the `<SignatureRequest>`. Any changes to the `<S12:Body>` will result in the `<SignatureRequest>` becoming invalid and therefore the SOAP Message Responder will detect the loss of integrity.

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

```

<?xml version="1.0" encoding="UTF-8"?>
<S12:Envelope xmlns:wsnr="http://schemas.reactivity.com/2003/04/wsnr/"
  xmlns:S12="http://www.w3.org/2002/12/soap-envelope"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/xx/utility"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/xx/secext">
  <S12:Header>
    <wsse:Security>
      <wsse:BinarySecurityToken EncodingType="wsse:Base64Binary"
        wsu:Id="RequestorCert">
        MIEFzCCA+igAwIBAgIBAzANBgkqhkiG9w0BAQQFADCB3DELMAkGA1UEBhMCMVVMxEzARBgNVBAGT
        CkNhbGlmb3JuaWEuEDAOBgNVBACTB0JlbnG1vbnQxIDAeBgNVBAoTF1JlYWN0aXZpdHkgVGZzdCBD
        b21wYW55MS4wLAYDVQQLEyVSZWZjdG12aXR5IFRlc3QgQ2VydG1maWNhdGUgQXV0aG9yaXR5MS4w
        LAYDVQQDEyVSZWZjdG12aXR5IFRlc3QgQ2VydG1maWNhdGUgQXV0aG9yaXR5MS4wLAYDVQIYIwZD
        AQkBFhVzb21lb25lQHNvbWV3aGVyZS5jb20wHhcNMDIwODI1MDAyMzU5WbcNMDMwODI1MDAyMzU5
        WjCBnTELMAGAlUEBhMCMVVMxFjAUBgNVBAGTDTU1hc3NhY2hlc2V0dHMxDzANBgNVBACTBkVjc3Rv
        bjESMBAGAlUEBhMCMVVMxFjAUBgNVBAGTDTU1hc3NhY2hlc2V0dHMxDzANBgNVBACTBkVjc3Rv
        Q29tcGFueSBBMScwIgwYJKoZIhvcNAQkBFhVzb21lb25lQHNvbWV3aGVyZS5jb20wgZ8wDQYJKoZI
        hvcNAQEBBQADgY0AMIGJAoGBALM+RhnZwfT6s1vdFpn+amZ7CvJIIdmXJgCRBzvwczNqdJhFtwl4

```

```

548 NX7Po2YM7vn/nlHw0E3yP3cwKqfHfAzvls5TuEXnfvjQAgTvJZYudRoc+D1w2QBjCtg/ox/0WNC
549 wU9eiHuHC3fm5ewCsx/H0WwuIThpOyUbWSl1NFkCJoXBAGMBAAGjggGMMIIBiDAJBgNVHRMEAjAA
550 MCwGCWGSAGG+EIBDQQfFh1PcGVuU1NMIEdlbnVYXRlZCBkZlZ0aWZpY2F0ZTAdBgNVHQ4EFgQU
551 PvkJwoTrduf/QbKxmPPZRGplls8wggEKBgNVHSMGggEBMIH+gBRsm+Jodl091efBrp8LkN/UC76N
552 AqGB4qSB3zCB3DELMaKGA1UEBhMCMVVMxEZARBgNVBAGTCkNhbG1mb3JuaWEwEwEADAQOBgNVBAcTB0Jl
553 bG1vbnQxIDAeBgNVBAoTF1JlYWN0aXZpdHkgVGVzdCBDb21wYW55MS4wLAYDVQQLLEyVSZWFjdG12
554 aXR5IFRlc3QgQ2VydG1maWNhdGUgQXV0aG9yaXR5MS4wLAYDVQDEyVSZWFjdG12aXR5IFRlc3Qg
555 Q2VydG1maWNhdGUgQXV0aG9yaXR5MSQwIgwYJKoZIhvcNAQkBFhVzb21lb25lQHNvbWV3aGVyZS5j
556 b22CAQAwCwYDVR0PBAQDAgWgMBMGALUdJQQMMAoGCCsGAQUFBwMCMA0GCsGqSIB3DQEBBAUAA4GB
557 AHpycTmqU2cMnlk8lAEgG+WuD6zP5GWqBgdw199J3JuDpfg/1fiF1QhCQJi/53DYO/edogVt276n
558 2pPcWqoemRnhVjmsGe0GzkQHFP445/++g1RuvOkhXthh2GoGI8P3tzAlwo8F7syJRxsEntF2j08E
559 ZbzPUK1B+TuC3MsRk0gi</wsse:BinarySecurityToken>
560 <ds:Signature>
561 <ds:SignedInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
562 <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-
563 c14n-20010315"/>
564 <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
565 sha1"/>
566 <ds:Reference URI="#receiptRequest2328348">
567 <ds:Transforms>
568 <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
569 20010315"/>
570 </ds:Transforms>
571 <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
572 <ds:DigestValue>w1TM8NLGwt6ZAuM/yX1Cu/1gv3I=</ds:DigestValue>
573 </ds:Reference>
574 </ds:SignedInfo>
575
576 <ds:SignatureValue>ekkILVSMIageKDVk5hNpD8F6QBfNlRY5bwsS41Q/VLiIQxPlULCGHHFIM68
577 uQMKW2E7wQ9ohiQe
578 x3aykuRT5HntpA9BI31EP2BPSlqOfjl61iMzKhKHQxYXlixsg8CdglCjaAylPDxCQoskF1cgjHOr
579 U6E7d9Ag9s33HqKGX2Q=</ds:SignatureValue>
580 <ds:KeyInfo>
581 <wsse:SecurityTokenReference>
582 <wsse:Reference URI="#RequestorCert"/>
583 </wsse:SecurityTokenReference>
584 </ds:KeyInfo>
585 </ds:Signature>
586 <wsnr:ReceiptRequest ReceiptFormat="signedReceipt" S12:mustUnderstand="true">
587 wsnr:CorrelationId="urn:uuid:f81d4fde-7dec-11d0-a765-00a0c91e6bf6"
588 wsu:Id="receiptRequest2328348">
589 <wsnr:ReceiptTo/>
590 <wsnr:SignatureRequest>
591 <ds:SignedInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
592 <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-
593 xml-c14n-20010315"/>
594 <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
595 sha1"/>
596 <ds:Reference URI="#body2328348">
597 <ds:Transforms>
598 <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
599 20010315"/>
600 </ds:Transforms>
601 <ds:DigestMethod
602 Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
603 <ds:DigestValue>9bKI84lheW6NCbnjltD4ZJi0wZ0=</ds:DigestValue>
604 </ds:Reference>
605 <ds:Reference URI="#timestamp2328348">
606 <ds:Transforms>
607 <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
608 20010315"/>
609 </ds:Transforms>
610 <ds:DigestMethod
611 Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
612 <ds:DigestValue>uNFkh+T9hVqvjKhmt61Gc90jMkI=</ds:DigestValue>
613 </ds:Reference>
614 </ds:SignedInfo>
615 </wsnr:SignatureRequest>
616 <wsu:Timestamp wsu:Id="timestamp2328348">
617 <wsu:Created>2003-03-12</wsu:Created>
618 </wsu:Timestamp>

```

```

619     </wsnr:ReceiptRequest>
620     </wsse:Security>
621   </S12:Header>
622   <S12:Body wsu:Id="body2328348">
623     <getTemperature xmlns="http://tempuri.org/temperature">
624       <city xsi:type="xsd:string">San Francisco</city>
625       <state xsi:type="xsd:string">CA</state>
626       <scale xsi:type="xsd:string">Celsius</scale>
627     </getTemperature>
628   </S12:Body>
629 </S12:Envelope>

```

630 10.3.2 Response

631 The SOAP Message Responder generates a **<SignatureResponse>** and includes it in a
632 **<Receipt>**. Then both the **<Receipt>** and the **<S12:Body>** are signed together.

```

633 <?xml version="1.0" encoding="UTF-8"?>
634 <S12:Envelope xmlns:wsnr="http://schemas.reactivity.com/2003/04/wsnr/"
635 xmlns:S12="http://www.w3.org/2002/12/soap-envelope"
636 xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
637 xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/xx/utility"
638 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
639 xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/xx/secext">
640   <S12:Header>
641     <wsse:Security>
642       <wsse:BinarySecurityToken EncodingType="wsse:Base64Binary"
643 wsu:Id="ResponderCert">
644 MIEfTCCA+agAwIBAgIBBDANBgkqhkiG9w0BAQQFADCB3DELMAkGA1UEBhMCVVMxEzARBgNVBAgT
645 CkNhbG1mb3JuaWEExEDAOBgNVBAcTB0JlbG1vbnQxIDAeBgNVBAoTF1JlYWNoaXZpdHkgVGVzdCBD
646 b21wYW5MS4wLAYDVQQLLEyVSZWFjdG12aXR5IFRlc3QgQ2VydG1maWNhdGUgQXV0aG9yaXR5MS4w
647 LAYDVQQDEyVSZWFjdG12aXR5IFRlc3QgQ2VydG1maWNhdGUgQXV0aG9yaXR5MS4wLAYDVQQLLEy
648 AQkBFhVzb211b251QHNVbWV3aGVyZS5jb20wHhcNMDIwODI1MDAzMzAzWhcNMDMwODI1MDAzMzAz
649 WjCBmzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1N1YXR0bGUx
650 EjAQBgNVBAoTCUNvbnBhbnkgQjEXMBUGA1UECXMOT3JnYW5pemF0aW9uIEIxEjAQBgNVBAMTCUNv
651 bXBhbnkgQjEkMCIGCSqGSIb3DQEJARIVVc29tZW9uZUBzb211d2h1cmUuY29tMIGfMA0GCSqGSIb3
652 DQEBAQUAA4GNADCBiQKBggQDc38GrOt/UYJZ8X+IbFlaxTZiwsFYpaztru7bQrDrx9sVcD9j3q6e
653 xl/iILkXhQzJ1tm9DEo+9VpNSTuCLhms5MHVdpFxsJlapXyv9P4AkyZFW/jiXx7AwP4nCTW4/6
654 XAOAuhQ0FJqemNUGwc51Y021X1NxQ/gb+6ggwSOZpwIDAQABo4IBjDCCAYgwCQYDVR0TBAIwADAs
655 BglghkgBhvhCAQ0EHEXydT3B1b1N1b1NTTCBHZW51cmF0ZWQgQ2VydG1maWNhdGUwHQYDVR0OBBYEFDjg
656 OM25FSBY3dP/9RUKIUWALqEUMIIBCgYDVDR0jBIIBATCB/oaUbjviahZTvdXnwa6fC5Df1Au+jQKh
657 geKkgd8wgdwxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEWpDYWxpZm9ybmlhMRAwDgYDVQQHEWdCZWxt
658 b250MSAwHgYDVQQKEXdSZWZjdG12aXR5IFRlc3QgQ29tZW9uZUBzb211d2h1cmUuY29tMIGfMA0G
659 eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1
660 cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1
661 IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0
662 eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UE
663 AxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpd
664 ml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1
665 cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IE
666 F1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTE
667 uMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1
668 UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0e
669 SBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnR
670 pm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1d
671 Ghvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuM
672 CwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1Um
673 VhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eS
674 BUZXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnR
675 pm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1d
676 Ghvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuM
677 CwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1Um
678 VhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eS
679 BUZXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnR
680 pm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1d
681 Ghvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuM
682 CwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1Um
683 VhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eS
684 BUZXN0IEN1cnRpm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnR
685 pm1jYXR1IEF1dGhvcml0eTEuMCwGA1UEAxM1UmVhY3Rpdml0eSBUXN0IEN1cnRpm1jYXR1IEF1d

```

686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725

```
</ds:Reference>
</ds:SignedInfo>

<ds:SignatureValue>C+5+owrA/c36aUJ3gGpCOJpy93/ueFm+eTM6ePFpKT65y23qUX00XNfF2IQ4
cS6HcUJUzVlp3ghD
fwZw4kVcgTgMWQLaEr7PwURME7ubzyxlepHDF0M4ysxEJsJ1NCzUAN8tIFXF7Ba4ganBhCaUOZm8
3GjtRRaqmRbi4sZuyU=</ds:SignatureValue>
<ds:KeyInfo>
  <wsse:SecurityTokenReference>
    <wsse:Reference URI="#ResponderCert"/>
  </wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
<wsnr:Receipt ReceiptFormat="signedReceipt"
wsnr:CorrelationId="urn:uuid:f81d4fde-7dec-11d0-a765-00a0c91e6bf6"
wsu:Id="receipt2328349">
  <wsnr:SignatureResponse>

<ds:SignatureValue>aaaWCUNlYJr/saEYyCP3PBaycNwP2w9rWqPNIdVRYV8tza5okFqlyJE9kB+k
xWovVoZItAQ+y/3R
xoSsGIwfdxZ3oUPxBsVJvPOOtrpZDVzGLT1cM2wQebcpurJZtt4yLQz6PP/cK2jcnJHUBHijmCa
wbWqZ3+V8o+6p97j+PI=</ds:SignatureValue>
<ds:KeyInfo>
  <wsse:SecurityTokenReference>
    <wsse:Reference URI="#ResponderCert"/>
  </wsse:SecurityTokenReference>
</ds:KeyInfo>
</wsnr:SignatureResponse>
<wsu:Timestamp wsu:Id="timestamp2328349">
  <wsu:Received>2003-03-12</wsu:Received>
</wsu:Timestamp>
</wsnr:Receipt>
</wsse:Security>
</S12:Header>
<S12:Body wsu:Id="body2328349">
  <getTemperatureResponse xmlns="http://tempuri.org/temperature">
    <temperature xsi:type="xsd:float">18.45</temperature>
  </getTemperatureResponse>
</S12:Body>
</S12:Envelope>
```

11 References

726

727

728
729
730
731
732
733
734
735
736
737
738
739
740

11.1 Normative

[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.

[UUID] M. Mealling, P. Leach, R. Salz. *A UUID URN Namespace*, <http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-00.txt>, IETF Internet-Draft, October 2002.

[SOAP11] W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.

[WSS] Web Services Security: SOAP Message Security
See: Oasis Web Services Security page: <http://www.oasis-open.org/committees/wss/>

[XMLDSIG] W3C Recommendation, "XML Signature Syntax and Processing," 12 February 2002.

[XMLENC] W3C Recommendation, "XML Encryption Syntax and Processing," 12 December 2002.

741 **11.2 Non-Normative**

742 **[Schneier]** B. Schneier, J. Kelsey. "Cryptographic Support for Secure Logs on
743 Untrusted Machines," Counterpane Systems, 23 October 1999
744 (<http://www.counterpane.com/secure-logs.pdf>).

745

Appendix A. Revision History

Rev	Date	By Whom	What
wd-01	2003-03-05	Eric Gravengaard	Initial version
wd-02	2003-03-10	Eric Gravengaard	Updated with comments from Grant and meeting on 3/6/2003
wd-03	2003-03-12	Eric Gravengaard	More updates and example.
wd-04	2003-04-01	Eric Gravengaard	Corrections and clarifications

746

747

Appendix B. Notices

748 Copyright © 2003 Reactivity, Inc. [REACTIVITY]. All Rights Reserved.

749 The furnishing of this specification does not grant you any rights or licenses, either expressly or
750 by implication, in any intellectual property owned or controlled by Reactivity or any other party,
751 whether necessary to implement the specification or otherwise. This document and the
752 information contained herein is provided on an "AS IS" basis and REACTIVITY DISCLAIMS ALL
753 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY
754 THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY
755 IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR
756 PURPOSE.