



LABORATORIES

# OTP-WSS-Token: Web Services Security One-Time-Password (OTP) Token Profile

## OASIS Submission Version, October 2005

### Document identifier:

Rsa-otps-otp-wss-token-v1-0

### Editor:

John Linn, RSA Laboratories <jlinn@rsasecurity.com>

### Abstract:

This document defines means to integrate multiple types of one-time password token-based authentication methods with Web Services Security, as defined in the OASIS "Web Services Security: SOAP Message Security 1.0" specification [WS-Security]. Many of the functions described within this document are similar and comparable to those defined in the OASIS "Web Services Security: UsernameToken Profile" [WS-UsernameToken], but the approaches described here have been tailored to provide more comprehensive support for authenticator token capabilities.

### Status:

**A predecessor version of this document [RSAS05a] was published by RSA Security Inc. within the One-Time Password Specifications (OTPS) series, to define an interoperable means to use token-based one-time password authentication within a web service framework. This document is being submitted by RSA Security as an input document to the OASIS Web Services Security (WSS) Technical Committee (TC) in order to support work on an OTP profile within that TC. RSA Security provides this submission in accordance with the provisions of the OASIS Legacy Intellectual Property Rights (IPR) Policy.**

---

28 **Table of Contents**

29 1 Introduction ..... 3  
30 2 Notation and Terminology..... 4  
31 2.1 Terminology..... 4  
32 2.2 Namespaces ..... 4  
33 3 OTPToken Data and Operations ..... 5  
34 3.1 Contents and Processing ..... 5  
35 3.2 Exception Cases ..... 7  
36 3.3 Token Placement and Referencing..... 8  
37 4 Security Considerations ..... 9  
38 5 Identifiers ..... 10  
39 6 Schema ..... 11  
40 7 References..... 13  
41 7.1 Normative ..... 13  
42 7.2 Non-normative..... 13  
43 Appendix A. Acknowledgments ..... 14  
44 Appendix B. Notices ..... 15  
45

---

## 46 1 Introduction

47 This document defines means to use authenticator token devices within the context of Web  
48 Services Security: SOAP Message Security (WSS:SMS). It defines an `<otps-wst:OTPToken>`  
49 element, which may be included as a custom security token within a Web Services Security  
50 message in order to authenticate the message's originator.

51

52 The profile defined in this document is intended to support use of authenticator token devices in  
53 three basic modes, or combinations thereof:

- 54 • Time-based mode, in which the token device generates a result value as a function of an  
55 internal clock and other internal data, without external inputs. This corresponds to the  
56 usage mode traditionally associated with hand-held RSA SecurID® authenticator token  
57 devices.
- 58 • Counter mode, in which the token device generates a result value as a function of an  
59 internal counter and other internal data, without external inputs.
- 60 • Challenge-response mode, in which the token device generates its result as a function of  
61 an input challenge value as well as data maintained internal to the device. In the context  
62 of this profile, challenge data may either be generated locally at the client system with  
63 which the authenticator device is used or may be received from the verifier system using  
64 a mechanism that is out-of-band relative to the protocol that this document defines. In  
65 either case, the challenge data is forwarded to the verifier system along with the result  
66 value obtained from the token.

67

---

68 **2 Notation and Terminology**

69 **2.1 Terminology**

70 The key words “MUST”, “RECOMMENDED”, and “MAY”, in this document are, when represented  
71 in upper case, to be interpreted as described in [RFC2119].

72 Common security terms are used as defined in the Internet Security Glossary [RFC2828].

73 **2.2 Namespaces**

74 This document uses the following namespaces:

Prefix	Namespace
otps-wst	<a href="http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#">http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
wsu	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd</a>

75

76

## 3 OTPToken Data and Operations

77

### 3.1 Contents and Processing

78

We define the `<otps-wst:OTPToken>` as a means to carry from a client to a validating server a set of data items that the server can process to validate an authentication based on a token device. The set of items that occur in a particular `OTPToken` will depend on the OTP method and profile being used, and on the characteristics of the specific authentication to be processed. `OTPToken` elements include:

82

83

- An optional `<otps-wst:TokTimestamp>` value, providing a timestamp value that is passed to the token device or obtained from it. For time-based token algorithms, this element can be used to carry the time value that is used in the OTP generation process. This element is not intended to convey the time(s) at which the `<otps-wst:OTPToken>` object is formatted or transmitted, should these differ from the time used in OTP generation. As such, it is not intended for use as a message-level replay detection facility, though may be used to detect replay at the OTP processing layer. Its optional `TSApprox` attribute can carry a Boolean value indicating (if true) that the timestamp is approximate, in that it may not exactly match the value used in the OTP computation though should be close to it; verifiers receiving a timestamp so indicated may wish to perform trials with a range of values close to the timestamp provided.

84

85

86

87

88

89

90

91

92

93

94

- An optional `<otps-wst:TokNonce>` value, providing a nonce value to the token device. For challenge-based token algorithms, this element can carry a challenge value (whether obtained from a verifier or generated locally at a client) used as input to OTP processing. For time-based token algorithms, the element can be used to ensure generation of unique results even when multiple authentications are performed within the time quantum of the `<otps-wst:TokTimestamp>` value. Similarly to `<otps-wst:TokTimestamp>`, it is intended to reflect a value used and managed at the OTP processing layer rather than for message-level processing purposes; it is not intended for use as a message-level replay detection facility though may be used to detect replay at the OTP processing layer.

95

96

97

98

99

100

101

102

103

104

- An optional `<otps-wst:TokCounter>` value, For counter-based token algorithms, this element can carry the counter value used in the OTP generation process. The counter value is a non-negative integer.

105

106

107

- An optional `<otps-wst:TokState>` value, carrying additional state information transferred from the token device to the validating server in order to support validation of the `<otps-wst:OTPToken>`.

108

109

110

- An optional PIN value `<otps-wst:TokPIN>` provided by the token device's user to be transferred to the validating server. Note that two types of PINs may be relevant for various types of token devices: a PIN obtained and processed locally within the token device in order to unlock it, and a PIN that is transferred along with an OTP value to a server in order to perform or validate an authentication. The `<otps-wst:TokPIN>` is intended for use with the second type of PIN, not the first.

111

112

113

114

115

116

- An optional `<otps-wst:ServID>` indicator, providing the token device with a character string identifier of the service with which the token's result is to be used.

117

118

- The result `<otps-wst:OTP>` being transferred to the validating server in order to perform OTP-based authentication. Certain OTP methods may include a user PIN as part of the `<otps-wst:OTP>`, additionally or alternatively to inclusion in the `<otps-`

119

120

121           wst:TokPIN> element; in these cases, PIN characters must be distinguished from other  
122 values within the <otps-wst:OTP> through means outside the scope of this  
123 specification.

124 The <otps-wst:OTPToken> can carry the following attributes:

- 125       • The TokQual attribute group allows a TokUser attribute (representing the claimed identity  
126 of a token device user) and/or a KeyID attribute (identifying the key used for OTP  
127 generation) to be provided. While each of these attributes is individually optional at the  
128 syntactic level, at least one must be present in order to comprise a semantically valid  
129 <otps-wst:OTPToken> object. As noted in [RSAS05], Sec. 3.8.5, key identifiers are  
130 expected to be globally unique. As one means to satisfy this requirement, that document  
131 suggests that the combination of an issuer's name and an issuer-specific unique serial  
132 number be hashed in order to produce a unique value with high probability.
- 133       • A wsu:Id attribute allows an identifier to be associated with a particular instance of an  
134 <otps-wst:OTPToken>.
- 135       • The TokAlg attribute provides a URI that identifies the type of cryptographic processing  
136 performed by the token device, and that qualifies interpretation and use of elements and  
137 attributes within the <otps-wst:OTPToken>. Values initially defined within this  
138 document distinguish the original RSA SecurID ALGOR (#SecurID-ALGOR) algorithm  
139 from the newer RSA SecurID AES algorithm (#SecurID-AES). (These URI fragments are  
140 relative to this specification's base URI, defined as "otps-wst" in Section 2.2 of this  
141 document.) Additional values may be defined in later versions of this document, or in  
142 other specifications. This attribute has no default value. The attribute must be included,  
143 and must specify a value, except in environments where the corresponding URI can  
144 otherwise be determined without ambiguity. In many cases, verifiers will be able to make  
145 such a determination based on the contents of TokUser and/or KeyID attributes.
- 146       • The TokOTPTransform attribute identifies a preprocessing algorithm that is applied to the  
147 OTP value as generated by the token device before inclusion in the <otps-wst:OTP>; if  
148 omitted, no preprocessing is performed. As one example of usage for this attribute, it  
149 could be used to specify that base64 encoding has been applied to represent an OTP  
150 value emitted as binary data by a connected token device.

151

152 It is RECOMMENDED that an <otps-wst:OTPToken> be encrypted in transit, to prevent  
153 possible interception and use of its <otps-wst:OTP> by unauthorized intermediaries. In addition  
154 to OTP values themselves, which may be subject to interception and unauthorized use  
155 (particularly if delivery of the original OTPToken to its verifier can be suppressed by an attacker),  
156 an <otps-wst:OTPToken> may also carry other sensitive data such as a PIN. Depending on  
157 operational context, an <otps-wst:OTPToken> may be encrypted at the XML layer or within a  
158 lower-layer encapsulation such as SSL or TLS.

159 If no <otps-wst:TokTimestamp> is provided, and a time-based token algorithm is in use, use  
160 of a time-based value derived within the token device is assumed. When a <otps-  
161 wst:TokTimestamp> is provided, senders MAY also provide a <otps-wst:TokNonce> and  
162 MUST ensure that the combination of <otps-wst:TokTimestamp> and <otps-  
163 wst:TokNonce> (if provided) is unique each time that generation of a <otps-wst:OTP> is  
164 requested. The <otps-wst:TokTimestamp> and <otps-wst:TokNonce> elements may  
165 either be generated at the client system or may be obtained from a verifier through means  
166 unspecified within this specification; when verifier-provided elements are employed, clients MAY  
167 satisfy the uniqueness obligation by relying on their associated verifiers to provide unique values  
168 for each instance.

169 In all cases, additional data items may also be maintained within the token device and may be  
170 incorporated in the overall cryptographic constructions that it implements.

171 A concrete example of an `<otps-wst:OTPToken>` is as follows:

```
172 <otps-wst:OTPToken wsu:Id="AnExampleOTPToken"  
173 TokAlg="http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-  
174 wst#SecurID-AES" TokUser="J. Sample User">  
175 <otps-wst:TokTimestamp>2005-01-09T20:25:42Z</otps-wst:TokTimestamp>  
176 <otps-wst:TokNonce>NIua4LqKeq3ciFzEv/MbZhA=</otps-wst:TokNonce>  
177 <otps-wst:TokPIN>876123</otps-wst:TokPIN>  
178 <otps-wst:OTP>142857</otps-wst:OTP>  
179 </otps-wst:OTPToken>
```

180

## 181 3.2 Exception Cases

182 No SOAP fault should be returned to a requester when an `<otps-wst:OTPToken>` is  
183 successfully validated. When an `<otps-wst:OTPToken>` cannot be successfully validated, but  
184 other aspects of SOAP processing have succeeded, a SOAP fault may be returned carrying a  
185 `Fault/Code/Subcode/Value` of `wsse:FailedAuthentication`.

186 When `wsse:FailedAuthentication` is indicated, the verifier can indicate the need for the  
187 requester to provide a new PIN by also including a `Fault/Detail` entry of `otps-wst:NeedNewPIN`  
188 within the returned fault. If such an entry is received, the requester should contact the verifier's  
189 PIN change service, providing a new PIN. Following this transaction, it should then generate a  
190 new `<otps-wst:OTPToken>` and retry its Web Services Security request in an attempt to re-  
191 authenticate. The PIN change service will be specified separately from this document. Within an  
192 `otps-wst:NeedNewPIN` entry, an information item MAY provide a URL for the verifier's PIN  
193 change service; alternately, its location and access method can be determined through  
194 configuration data and/or metadata exchange.

195 When token devices are operated in time-based or counter-based mode, it is possible for a  
196 verifier to recognize that a particular `<otps-wst:OTP>` lies within the acceptable window, but  
197 that it corresponds to a time or counter value far enough from the expected value that an  
198 additional token result must be obtained and verified before an authentication can be accepted.  
199 A verifier can indicate this contingency with `wsse:FailedAuthentication` in conjunction with  
200 a `Fault/Detail` entry of `otps-wst:NextTokencode` within the returned fault. If such an entry is  
201 received, the requester should retry its Web Services Security request, providing the next result  
202 value obtained from the token device in the `<otps-wst:OTP>` element, and attempt to re-  
203 authenticate. External orchestration facilities outside the scope of this specification must be used  
204 to recognize and process cases such as these, when an authentication request is related to and  
205 supplements a prior request.

206 When token devices are operated in time-based mode, a verifier may recognize that the time  
207 value represented in a particular `<otps-wst:OTPToken>` lies outside the acceptable window. A  
208 verifier can indicate this contingency with `wsse:FailedAuthentication` in conjunction with a  
209 `Fault/Detail` entry of `otps-wst:WrongTime` within the returned fault. If such an entry is received,  
210 the requester should attempt to correct its clock setting before retrying its Web Services Security  
211 request.

212 When token devices are operated in counter-based mode, a verifier may recognize that the  
213 counter value represented in a particular `<otps-wst:OTPToken>` lies outside the acceptable  
214 window. A verifier can indicate this contingency with `wsse:FailedAuthentication` in  
215 conjunction with a `Fault/Detail` entry of `otps-wst:WrongCounter` within the returned fault. If  
216 such an entry is received, the requester should attempt to correct its counter value before retrying  
217 its Web Services Security request.

218 When verifiers provide challenges to be used as input to generation of an OTP value, it is  
219 possible that a requester will provide an `<otps-wst:OTPToken>` with an incorrect challenge. A

220 verifier can indicate this contingency with `wsse:FailedAuthentication` in conjunction with a  
221 Fault/Detail entry of `otps-wst:WrongChallenge` within the returned fault. If such an entry is  
222 received, the requester should obtain a fresh challenge from the verifier and use the resulting  
223 OTP value to initiate a new request.

### 224 **3.3 Token Placement and Referencing**

225 When used in WSS:SMS environments, it is recommended in the interests of interoperability that  
226 any `<otps-wst:OTPToken>` be included as an immediate descendant of the enclosing  
227 `<wsse:Security>` header, without `<wsse:Embedded>` wrapping.

228

229 The ValueType `#OTPToken` may be used to qualify references to an `<otps-wst:OTPToken>`.

---

## 230 4 Security Considerations

231 Security considerations applicable to the use of an `<otps-wst:OTPToken>` are generally similar  
232 to those of other token types used with Web Services Security. Encryption of the `<otps-`  
233 `wst:OTPToken>` itself, or use of a cryptographically secure channel to protect its transport, is  
234 strongly recommended in order to avoid interception and misuse of the `<otps-wst:OTP>` within  
235 the `<otps-wst:OTPToken>`.

236

237 When token devices are operated in challenge-response mode, it is necessary for the generators  
238 of `<otps-wst:TokTimestamp>` and `<otps-wst:TokNonce>` data to ensure that the  
239 combination of their values are unique each time a `<otps-wst:OTP>` is to be generated, so as  
240 not to inadvertently trigger verifier-side replay detection mechanisms.

---

## 241 5 Identifiers

242 This document defines the following identifiers for token device algorithms via URIs:

- 243 • The URI [http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#SecurID-](http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#SecurID-AES)  
244 [AES](http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#SecurID-AES) indicates the use of the SecurID-AES processing algorithm.
- 245 • The URI [http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#SecurID-](http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#SecurID-ALGOR)  
246 [ALGOR](http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#SecurID-ALGOR) indicates the use of the SecurID-ALGOR processing algorithm.

247 The URI <http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#OTPToken>  
248 provides a ValueType to qualify references to OTPTokens within enclosing XML documents.

249 The following values are defined for use in SOAP fault/detail entries to report specific OTP-  
250 related events, when provided in conjunction with an overall `wsse:FailedAuthentication`  
251 faultcode:

- 252 • The URI [http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-](http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#NeedNewPIN)  
253 [wst#NeedNewPIN](http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#NeedNewPIN) indicates that a new PIN must be provided before authentication can  
254 be completed.
- 255 • The URI [http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-](http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#NextTokencode)  
256 [wst#NextTokencode](http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#NextTokencode) indicates that a token device's next OTP value must be provided  
257 before authentication can be completed.
- 258 • The URI [http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-](http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#WrongChallenge)  
259 [wst#WrongChallenge](http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#WrongChallenge) indicates that an unexpected challenge value was processed within  
260 an OTPToken and that a new challenge must be obtained and used to generate a new  
261 OTP value before authentication can be completed.
- 262 • The URI <http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#WrongTime>  
263 indicates that an inappropriate time value was processed within an OTPToken and that  
264 the clock must be corrected and used to generate a new OTP value before authentication  
265 can be completed.
- 266 • The URI [http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-](http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#WrongCounter)  
267 [wst#WrongCounter](http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-wst#WrongCounter) indicates that an unexpected counter value was processed within an  
268 OTPToken and that the counter value must be corrected and used to generate a new  
269 OTP value before authentication can be completed.

270

271

272

## 6 Schema

```
274 <?xml version="1.0" encoding="UTF-8"?>
275 <!--
276 Copyright (c) RSA Security Inc. 2005. All rights reserved.
277 -->
278
279 <xs:schema
280 targetNamespace="http://www.rsasecurity.com/rsalabs/otps/schemas/2005/0
281 9/otps-wst#"
282 xmlns="http://www.rsasecurity.com/rsalabs/otps/schemas/2005/09/otps-
283 wst#"
284 xmlns:xs="http://www.w3.org/2001/XMLSchema"
285 xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
286 wssecurity-utility-1.0.xsd"
287 elementFormDefault="qualified"
288 attributeFormDefault="unqualified"
289 version="1.0"
290 id="OTPS-WSS-Token">
291
292 <xs:import namespace="http://docs.oasis-open.org/wss/2004/01/oasis-
293 200401-wss-wssecurity-utility-1.0.xsd"
294 schemaLocation="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
295 wss-wssecurity-utility-1.0.xsd"/>
296
297 <xs:attributeGroup name="TokQual">
298   <xs:attribute name="TokUser" type="xs:string" use="optional"/>
299   <xs:attribute name="KeyID" type="xs:base64Binary" use="optional"/>
300 </xs:attributeGroup>
301
302 <xs:complexType name="OTPToken">
303   <xs:annotation>
304     <xs:documentation>
305 Type definition for token-based authentication
306     </xs:documentation>
307   </xs:annotation>
308   <xs:sequence>
309     <xs:element name="TokTimestamp" minOccurs="0">
310       <xs:complexType>
311         <xs:simpleContent>
312           <xs:extension base="xs:dateTime">
313             <xs:attribute name="TSApprox"
314 type="xs:boolean" use="optional"/>
315           </xs:extension>
316         </xs:simpleContent>
317       </xs:complexType>
318     </xs:element>
319     <xs:element name="TokNonce" type="xs:base64Binary"
320 minOccurs="0"/>
321     <xs:element name="TokCounter" type="xs:nonNegativeInteger"
322 minOccurs="0"/>
323     <xs:element name="TokState" type="xs:base64Binary"
324 minOccurs="0"/>
325     <xs:element name="TokPIN" type="xs:string" minOccurs="0"/>
326     <xs:element name="ServID" type="xs:string" minOccurs="0"/>
327     <xs:element name="OTP" type="xs:string"/>
328     <xs:any minOccurs="0" maxOccurs="unbounded"/>
329   </xs:sequence>
```

```
330 <xs:attributeGroup ref="TokQual"/>
331 <xs:attribute ref="wsu:Id" use="optional"/>
332 <xs:attribute name="TokAlg" type="xs:anyURI" use="optional"/>
333 <xs:attribute name="TokOTPTransform" type="xs:anyURI"
334 use="optional"/>
335 </xs:complexType>
336
337 <xs:complexType name="NeedNewPIN">
338 <xs:annotation>
339 <xs:documentation>
340 Info for New PIN request, used in SOAP fault's env:detail ##any
341 </xs:documentation>
342 </xs:annotation>
343 <xs:sequence>
344 <xs:element name="PINChgSvc" type="xs:anyURI" minOccurs="0"/>
345 </xs:sequence>
346 </xs:complexType>
347
348 </xs:schema>
```

349

---

350 **7 References**

351 **7.1 Normative**

- 352 [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,  
353 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 354 [RFC2828] R. Shirey, *Internet Security Glossary*, <http://www.ietf.org/rfc/rfc2828.txt>,  
355 IETF RFC 2828, May 2000.
- 356 [SOAP12] W3C Recommendation, "SOAP Version 1.2 Part 1: Messaging  
357 Framework", 24 June 2003.
- 358 [XML-Schema] W3C Recommendation, "XML Schema Part 1: Structures", 2 May 2001.  
359 W3C Recommendation, "XML Schema Part 2: Datatypes", 2 May 2001.
- 360 [WS-Security] OASIS, "Web Services Security: SOAP Message Security 1.0 (WS-  
361 Security 2004)", OASIS Standard, March 2004.  
362

363 **7.2 Non-normative**

- 364 [RSAS05] RSA Security, "Cryptographic Token Key Initialization Protocol", V1.0  
365 Draft 4, 26 August 2005.
- 366 [RSAS05a] **RSA Security, "OTP-WSS-Token: Web Services Security One-Time  
367 Password (OTP) Token Profile", Version 1-0, 22 September 2005.**
- 368
- 369 [WS-UsernameToken] OASIS, "Web Services Security: UsernameToken Profile 1.0",  
370 OASIS Standard, March 2004.  
371

---

372

## Appendix A. Acknowledgments

373 Thanks to the OASIS Web Services Security TC, and particularly to the editors and contributors  
374 to the Web Services Security: SOAP Message Security and Web Services Security:  
375 UsernameToken Profile specifications, for their work on the framework definitions that supported  
376 and provided parallels for this document.

377

378 Thanks to Piers Bowness and Magnus Nyström of RSA Security Inc., James Manger of Telstra,  
379 and Philip Hoyer of ActivCard (UK) for their contributions to drafts of this document. Thanks also  
380 to all of the participants at the May 2005 OTPS workshop and on the OTPS mailing list for their  
381 review and comments.

382

## Appendix B. Notices

383 This document defines means to use authenticator token devices, such as RSA SecurID®  
384 authenticators, within the context of Web Services Security as defined in the OASIS "Web  
385 Services Security: SOAP Message Security 1.0" specification. This document is intended to  
386 assist in the implementation of this WS-Security specification.  
387

388 Copyright © 2005 RSA Security Inc. All rights reserved. License to copy this document and  
389 furnish the copies to others is granted provided that the above copyright notice is included on all  
390 such copies.

391 RSA, RSA Security and SecurID are registered trademarks or trademarks of RSA Security Inc. in  
392 the United States and/or other countries. The names of other products or services mentioned  
393 may be the trademarks of their respective owners.  
394

395 The OASIS "Web Services Security: SOAP Message Security 1.0" specification is Copyright ©  
396 OASIS Open 2002-2004. *All Rights Reserved.*  
397

398 The following is reproduced from the specification: "This document and translations of it may be  
399 copied and furnished to others, and derivative works that comment on or otherwise explain it or  
400 assist in its implementation may be prepared, copied, published and distributed, in whole or in  
401 part, without restriction of any kind, provided that the above copyright notice and this paragraph  
402 are included on all such copies and derivative works. However, this document itself does not be  
403 modified in any way, such as by removing the copyright notice or references to OASIS, except as  
404 needed for the purpose of developing OASIS specifications, in which case the procedures for  
405 copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as  
406 required to translate it into languages other than English."  
407

408 RSA Security does not make any claims on the general constructions described in this document.  
409 Specific underlying methods and techniques that may be supported and represented using  
410 facilities defined in this document may be subject to claims. For example, the RSA SecurID  
411 technology implementations of time-based mode authenticator token devices, and related  
412 validation processing components, are covered by a number of US patents (and foreign  
413 counterparts), in particular US Patent Nos. 4,885,778; 4,856,062; 5,097,505; 5,168,520 and  
414 5,657,388. Additional patents are pending. As this specification can be implemented without the  
415 use of time-based mode authentication technology, it is RSA Security's position that the  
416 technology covered by these patents and applications is not required to implement this  
417 specification.

418 This document and the information contained herein are provided on an "AS IS" basis and RSA  
419 SECURITY DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT  
420 LIMITED TO, ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT  
421 INFRINGE ANY INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY, AND ANY  
422 IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR  
423 PURPOSE. In accordance with Section OASIS.IPR.3.1 of the Legacy OASIS Intellectual  
424 Property Rights (IPR) Policy, RSA Security represents that named contributors in Appendix A  
425 were made aware of and agreed to accept the terms and conditions enumerated in that section.  
426 RSA Security makes no other representations regarding intellectual property claims by other  
427 parties. Such determination is the responsibility of the user.

