



Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V1.1

Last Call Working Draft 02, 2 May 2003

Document identifier:

sstc-saml-sec-consider-1.1-draft-02

Location:

http://www.oasis-open.org/committees/documents.php?wg_abbrev=security

Editor:

Eve Maler, Sun Microsystems (eve.maler@sun.com)

Contributors:

Tim Moses, Entrust Inc.
Chris McLaren, Netegrity (former editor)
Prateek Mishra, Netegrity
Jeff Hodges, Sun Microsystems
Evan Prodromou, formerly of Securant
Marlena Erdos, Tivoli
RL "Bob" Morgan, University of Washington and Internet2

Abstract:

This specification describes and analyzes the security and privacy properties of SAML.

Status:

This document is a **last-call working draft** of the OASIS Security Services Technical Committee. We solicit your comments; they must be received by Friday, 16 May 2003 in order for the committee to consider them for inclusion in the Committee Specification.

If you are on the security-services@lists.oasis-open.org list for committee members, send comments there. If you are not on that list, subscribe to the security-services-comment@lists.oasis-open.org list and send comments there. To subscribe, send an email message to security-services-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Security Services TC web page (<http://www.oasis-open.org/committees/security/>).

Table of Contents

36	1	Introduction.....	4
37	2	Privacy.....	5
38	2.1	Ensuring Confidentiality.....	5
39	2.2	Notes on Anonymity.....	5
40	2.2.1	Definitions That Relate to Anonymity.....	5
41	2.2.2	Pseudonymity and Anonymity.....	6
42	2.2.3	Behavior and Anonymity.....	7
43	2.2.4	Implications for Privacy.....	7
44	3	Security.....	8
45	3.1	Background.....	8
46	3.2	Scope.....	8
47	3.3	SAML Threat Model.....	8
48	4	Security Techniques.....	10
49	4.1	Authentication.....	10
50	4.1.1	Active Session.....	10
51	4.1.2	Message-Level.....	10
52	4.2	Confidentiality.....	10
53	4.2.1	In Transit.....	10
54	4.2.2	Message-Level.....	10
55	4.3	Data Integrity.....	10
56	4.3.1	In Transit.....	11
57	4.3.2	Message-Level.....	11
58	4.4	Notes on Key Management.....	11
59	4.4.1	Access to the Key.....	11
60	4.4.2	Binding of Identity to Key.....	11
61	4.5	TLS/SSL Cipher Suites.....	12
62	4.5.1	What Is a Cipher Suite?.....	12
63	4.5.2	Cipher Suite Recommendations.....	13
64	5	SAML-Specific Security Considerations.....	14
65	5.1	SAML Assertions.....	14
66	5.2	SAML Protocol.....	14
67	5.2.1	Denial of Service.....	14
68	5.2.1.1	Requiring Client Authentication at a Lower Level.....	14
69	5.2.1.2	Requiring Signed Requests.....	15

70 5.2.1.3 Restricting Access to the Interaction URL..... 15

71 5.3 SAML Protocol Bindings 15

72 5.3.1 SOAP Binding..... 15

73 5.3.1.1 Eavesdropping.....15

74 5.3.1.2 Replay.....16

75 5.3.1.3 Message Insertion.....17

76 5.3.1.4 Message Deletion17

77 5.3.1.5 Message Modification17

78 5.3.1.6 Man-in-the-Middle18

79 5.3.2 Specifics of SOAP over HTTP..... 18

80 5.4 Profiles of SAML 18

81 5.4.1 Web Browser-Based Profiles 19

82 5.4.1.1 Eavesdropping.....19

83 5.4.1.1.1 Theft of the User Authentication Information19

84 5.4.1.1.2 Theft of the Bearer Token19

85 5.4.1.2 Replay.....20

86 5.4.1.3 Message Insertion.....20

87 5.4.1.4 Message Deletion20

88 5.4.1.5 Message Modification20

89 5.4.1.6 Man-in-the-Middle20

90 5.4.2 Browser/Artifact Profile..... 21

91 5.4.2.1 Replay.....21

92 5.4.3 Browser/POST Profile 21

93 5.4.3.1 Replay.....21

94 6 References 22

95 Appendix A. Acknowledgments 24

96 Appendix B. Notices 25

97 Appendix C. Revision History..... 26

98

1 Introduction

100 This non-normative document describes and analyzes the security and privacy properties of the OASIS
101 Security Assertion Markup Language (SAML) defined in the core SAML specification **[SAMLCore]** and
102 the SAML specification for bindings and profiles **[SAMLBind]**. The intent in this document is to provide
103 input to the design of SAML, and to provide information to architects, implementors, and reviewers of
104 SAML-based systems about the following:

- 105 • The threats, and thus security risks, to which a SAML-based system is subject
- 106 • The security risks the SAML architecture addresses, and how it does so
- 107 • The security risks it does not address
- 108 • Recommendations for countermeasures that mitigate those risks

109 Terms used in this document are as defined in the SAML glossary **[SAMLGloss]** unless otherwise noted.

110 The rest of this section describes the background and assumptions underlying the analysis in this
111 document. Section 4 provides a high-level view of security techniques and technologies that should be
112 used with SAML. Section 5 analyzes the specific risks inherent in the use of SAML.

113 2 Privacy

114 SAML includes the ability to make statements about the attributes and authorizations of authenticated
115 entities. There are very many common situations in which the information carried in these statements is
116 something that one or more of the parties to a communication would desire to keep accessible to as
117 restricted as possible a set of entities. Statements of medical or financial attributes are simple examples
118 of such cases.

119 Parties making statements, issuing assertions, conveying assertions, and consuming assertions must be
120 aware of these potential privacy concerns and should attempt to address them in their implementations of
121 SAML-aware systems.

122 2.1 Ensuring Confidentiality

123 Perhaps the most important aspect of ensuring privacy to parties in a SAML-enabled transaction is the
124 ability to carry out the transaction with a guarantee of confidentiality. In other words, can the information
125 in an assertion be conveyed from the issuer to the intended audience, and only the intended audience,
126 without making it accessible to any other parties?

127 It is technically possible to convey information confidentially (a discussion of common methods for
128 providing confidentiality occurs in the Security portion of the document in Section 4.2). All parties to
129 SAML-enabled transactions should analyze each of their steps in the interaction to ensure that
130 information that should be kept confidential is actually being kept so.

131 It should also be noted that simply obscuring the contents of assertions may not be adequate protection
132 of privacy. There are many cases where just the availability of the information that a given user (or IP
133 address) was accessing a given service may constitute a breach of privacy (for example, an the
134 information that a user accessed a medical testing facility for an assertion may be enough to breach
135 privacy without knowing the contents of the assertion). Partial solutions to these problems can be
136 provided by various techniques for anonymous interaction, outlined below.

137 2.2 Notes on Anonymity

138 The following sections discuss the concept of anonymity.

139 2.2.1 Definitions That Relate to Anonymity

140 There are no definitions of anonymity that are satisfying for all cases. Many definitions **[Anonymity]** deal
141 with the simple case of a sender and a message, and discuss “anonymity” in terms of not being able to
142 link a given sender to a sent message, or a message back to a sender.

143 And while that definition is adequate for the “one off” case, it ignores the aggregation of information that is
144 possible over time based on behavior rather than an identifier.

145 Two notions that may be generally useful, and that relate to each other, can help define anonymity.

146 The first notion is to think about anonymity as being “within a set”, as in this comment from “Anonymity,
147 Unobservability, and Pseudonymity” **[Anonymity]**:

148 To enable anonymity of a subject, there always has to be an appropriate set of subjects with
149 potentially the same attributes....

150 ...Anonymity is the stronger, the larger the respective anonymity set is and the more evenly
151 distributed the sending or receiving, respectively, of the subjects within that set is.

152 This notion is relevant to SAML because of the use of authorities. Even if a Subject is “anonymous”, that
153 subject is still identifiable as a member of the set of Subjects within the domain of the relevant authority.

154 In the case where aggregating attributes of the user are provided, the set can become much smaller – for
155 example, if the user is “anonymous” but has the attribute of “student in Course 6@mit.edu”. Certainly, the
156 number of Course 6 students is less than the number of MIT-affiliated persons which is less than the
157 number of users everywhere.

158 Why does this matter? Non-anonymity leads to the ability of an adversary to harm, as expressed in
159 Dingledine, Freedman, and Molnar’s Freehaven document [**FreeHaven**]:

160 Both anonymity and pseudonymity protect the privacy of the user’s location and true name.
161 Location refers to the actual physical connection to the system. The term “true name” was
162 introduced by Vinge and popularized by May to refer to the legal identity of an individual.
163 Knowing someone’s true name or location allows you to hurt him or her.

164 This leads to a unification of the notion of anonymity within a set and ability to harm, from the same
165 source [**FreeHaven**]:

166 We might say that a system is partially anonymous if an adversary can only narrow down a
167 search for a user to one of a ‘set of suspects.’ If the set is large enough, then it is impractical
168 for an adversary to act as if any single suspect were guilty. On the other hand, when the set of
169 suspects is small, mere suspicion may cause an adversary to take action against all of them.

170 SAML-enabled systems are limited to “partial anonymity” at best because of the use of authorities. An
171 entity about whom an assertion is made is already identifiable as one of the pool of entities in a
172 relationship with the issuing authority.

173 The limitations on anonymity can be much worse than simple authority association, depending on how
174 identifiers are employed, as reuse of pseudonymous identifiers allows accretion of potentially identifying
175 information (see Section 2.2.2). Additionally, users of SAML-enabled systems can also make the breach
176 of anonymity worse by their actions (see Section 2.2.3).

177 **2.2.2 Pseudonymity and Anonymity**

178 Apart from legal identity, any identifier for a Subject can be considered a pseudonym. And even notions
179 like “holder of key” can be considered as serving as the equivalent of a pseudonym in linking an action (or
180 set of actions) to a Subject. Even a description such as “the user that just requested access to object XYZ
181 at time 23:34” can serve as an equivalent of a pseudonym.

182 Thus, that with respect to “ability to harm,” it makes no difference whether the user is described with an
183 identifier or described by behavior (for example, use of a key or performance of an action).

184 What does make a difference is how often the particular equivalent of a pseudonym is used.

185 [**Anonymity**] gives a taxonomy of pseudonyms starting from personal pseudonyms (like nicknames) that
186 are used all the time, through various types of role pseudonyms (such as Secretary of Defense), on to
187 “one-time-use” pseudonyms.

188 Only one-time-use pseudonyms can give you anonymity (within SAML, consider this as “anonymity within
189 a set”).

190 The more often you use a given pseudonym, the more you reduce your anonymity and the more likely it is
191 that you can be harmed. In other words, reuse of a pseudonym allows additional potentially identifying
192 information to be associated with the pseudonym. Over time, this will lead to an accretion that can
193 uniquely identify the identity associated with a pseudonym.

194 **2.2.3 Behavior and Anonymity**

195 As Joe Klein can attest, anonymity isn't all it is cracked up to be.

196 Klein is the "Anonymous" who authored Primary Colors. Despite his denials he was unmasked as the
197 author by Don Foster, a Vassar professor who did a forensic analysis of the text of Primary Colors. Foster
198 compared that text with texts from a list of suspects that he devised based on their knowledge bases and
199 writing proclivities.

200 It was Klein's idiosyncratic usages that did him in (though apparently all authors have them).

201 The relevant point for SAML is that an "anonymous" user (even one that is never named) can be
202 identified enough to be harmed by repeated unusual behavior. Here are some examples:

- 203 • A user who each Tuesday at 21:00 access a database that correlates finger lengths and life span
204 starts to be non-anonymous. Depending on that user's other behavior, she or he may become
205 "traceable" [**Pooling**] in that other "identifying" information may be able to be collected.
- 206 • A user who routinely buys a usual set of products from a networked vending machine certainly opens
207 themselves to harm (by virtue of booby-trapping the products).

208 **2.2.4 Implications for Privacy**

209 Origin site authorities (such as authentication authorities and attribute authorities) can provide a degree of
210 "partial anonymity" by employing one-time-use identifiers or keys (for the "holder of key" case).

211 This anonymity is "partial" at best because the Subject is necessarily confined to the set of Subjects in a
212 relationship with the Authority.

213 This set may be further reduced (thus further reducing anonymity) when aggregating attributes are used
214 that further subset the user community at the origin site.

215 Users who truly care about anonymity must take care to disguise or avoid unusual patterns of behavior
216 that could serve to "de-anonymize" them over time.

217 3 Security

218 The following sections discuss security considerations.

219 3.1 Background

220 Communication between computer-based systems is subject to a variety of threats, and these threats
221 carry some level of associated risk. The nature of the risk depends on a host of factors, including the
222 nature of the communications, the nature of the communicating systems, the communication mediums,
223 the communication environment, the end-system environments, and so on. Section 3 of the IETF
224 guidelines on writing security considerations for RFCs [**Rescorla-Sec**] provides an overview of threats
225 inherent in the Internet (and, by implication, intranets).

226 SAML is intended to aid deployers in establishing security contexts for application-level computer-based
227 communications within or between security domains. By serving in this role, SAML addresses the
228 “endpoint authentication” aspect (in part, at least) of communications security, and also the “unauthorized
229 usage” aspect of systems security. Communications security is directly applicable to the design of SAML.
230 Systems security is of interest mostly in the context of SAML’s threat models. Section 2 of the IETF
231 guidelines gives an overview of communications security and systems security.

232 3.2 Scope

233 Some areas that impact broadly on the overall security of a system that uses SAML are explicitly outside
234 the scope of SAML. While this document does not address these areas, they should always be
235 considered when reviewing the security of a system. In particular, these issues are important, but
236 currently beyond the scope of SAML:

- 237 • Initial authentication: SAML allows statements to be made about acts of authentication that have
238 occurred, but includes no requirements or specifications for these acts of authentication. Consumers
239 of authentication assertions should be wary of blindly trusting these assertions unless and until they
240 know the basis on which they were made. Confidence in the assertions must never exceed the
241 confidence that the asserting party has correctly arrived at the conclusions asserted.
- 242 • Trust Model: In many cases, the security of a SAML conversation will depend on the underlying trust
243 model, which is typically based on a key management infrastructure (for example, PKI or secret key).
244 For example, SOAP messages secured by means of XML Signature [**XMLSig**] are secured only
245 insofar as the keys used in the exchange can be trusted. Undetected compromised keys or revoked
246 certificates, for example, could allow a breach of security. Even failure to require a certificate opens
247 the door for impersonation attacks. PKI setup is not trivial and must be implemented correctly in order
248 for layers built on top of it (such as parts of SAML) to be secure.

249 3.3 SAML Threat Model

250 The general Internet threat model described in the IETF guidelines for security considerations [**Rescorla-**
251 **Sec**] is the basis for the SAML threat model. We assume here that the two or more endpoints of a SAML
252 transaction are uncompromised, but that the attacker has complete control over the communications
253 channel.

254 Additionally, due to the nature of SAML as a multi-party authentication and authorization statement
255 protocol, cases must be considered where one or more of the parties in a legitimate SAML transaction—
256 who operate legitimately within their role for that transaction—attempt to use information gained from a
257 previous transaction maliciously in a subsequent transaction.

258 In all cases, the local mechanisms that systems will use to decide whether or not to generate assertions
259 are out of scope. Thus, threats arising from the details of the original login at an authentication authority,
260 for example, are out of scope as well. If an authority issues a false assertion, then the threats arising from
261 the consumption of that assertion by downstream systems are explicitly out of scope.

262 The direct consequence of such a scoping is that the security of a system based on assertions as inputs
263 is only as good as the security of the system used to generate those assertions. When determining what
264 issuers to trust, particularly in cases where the assertions will be used as inputs to authentication or
265 authorization decisions, the risk of security compromises arising from the consumption of false but validly
266 issued assertions is a large one. Trust policies between asserting and relying parties should always be
267 written to include significant consideration of liability and implementations must be provide an audit trail.

268 4 Security Techniques

269 The following sections describe security techniques and various stock technologies available for their
270 implementation in SAML deployments.

271 4.1 Authentication

272 Authentication here means the ability of a party to a transaction to determine the identity of the other party
273 in the transaction. This authentication may be in one direction or it may be bilateral.

274 4.1.1 Active Session

275 Non-persistent authentication is provided by the communications channel used to transport a SAML
276 message. This authentication may be unilateral—from the session initiator to the receiver—or bilateral.
277 The specific method will be determined by the communications protocol used. For instance, the use of a
278 secure network protocol, such as RFC 2246 [RFC2246] or the IP Security Protocol [IPsec], provides the
279 SAML message sender with the ability to authenticate the destination for the TCP/IP environment.

280 4.1.2 Message-Level

281 XML Signature [XMLSig] and the OASIS Web Services Security specifications [WSS] provide methods of
282 creating a persistent “authentication” that is tightly coupled to a document. This method does not
283 independently guarantee that the sender of the message is in fact that signer (and indeed, in many cases
284 where intermediaries are involved, this is explicitly not the case).

285 Any method that allows the persistent confirmation of the involvement of a uniquely resolvable entity with
286 a given subset of an XML message is sufficient to meet this requirement.

287 4.2 Confidentiality

288 Confidentiality means that the contents of a message can be read only by the desired recipients and not
289 anyone else who encounters the message.

290 4.2.1 In Transit

291 Use of a secure network protocol such as RFC 2246 [RFC2246] or the IP Security Protocol [IPsec]
292 provides transient confidentiality of a message as it is transferred between two nodes.

293 4.2.2 Message-Level

294 XML Encryption [XMLEnc] provides for the selective encryption of XML documents. This encryption
295 method provides persistent, selective confidentiality of elements within an XML message.

296 4.3 Data Integrity

297 Data integrity is the ability to confirm that a given message as received is unaltered from the version of
298 the message that was sent.

299 **4.3.1 In Transit**

300 Use of a secure network protocol such as RFC 2246 [**RFC2246**] or the IP Security Protocol [**IPsec**] may
301 be configured so as to provide for integrity check CRCs of the packets transmitted via the network
302 connection.

303 **4.3.2 Message-Level**

304 XML Signature [**XMLSig**] provides a method of creating a persistent guarantee of the unaltered nature of
305 a message that is tightly coupled to that message.

306 Any method that allows the persistent confirmation of the unaltered nature of a given subset of an XML
307 message is sufficient to meet this requirement.

308 **4.4 Notes on Key Management**

309 Many points in this document will refer to the ability of systems to provide authentication, data integrity,
310 and confidentiality via various schemes involving digital signature and encryption. For all these schemes
311 the security provided by the scheme is limited based on the key management systems that are in place.
312 Some specific limitations are detailed below.

313 **4.4.1 Access to the Key**

314 It is assumed that, if key-based systems are going to be used for authentication, data integrity, and non-
315 repudiation, security is in place to guarantee that access to the key is not available to inappropriate
316 parties. For example, a digital signature created with Bob's private key is only proof of Bob's involvement
317 to the extent that Bob is the only one with access to the key.

318 In general, access to keys should be kept to the minimum set of entities possible (particularly important
319 for corporate or organizational keys) and should be protected with passphrases and other means.
320 Standard security precautions (don't write down the passphrase, when you're away from a computer don't
321 leave a window with the key accessed open, and so on) apply.

322 **4.4.2 Binding of Identity to Key**

323 For a key-based system to be used for authentication there must be some trusted binding of identity to
324 key. Verifying a digital signature on a document can determine if the document is unaltered since it was
325 signed, and that it was actually signed by a given key. However, this in no way confirms that the key used
326 is actually the key of a specific individual.

327 This key-to-individual binding must be established. Common solutions include local directories that store
328 both identifiers and key—which is simple to understand but difficult to maintain—or the use of certificates.

329 Certificates, which are in essence signed bindings of identity-to-key are a particularly powerful solution to
330 the problem, but come with their own considerations. A set of trusted root Certifying Authorities (CAs)
331 must be identified for each consumer of signatures—answering the question “Whom do I trust to make
332 statements of identity-to-key binding?” Verification of a signature then becomes a process of verifying first
333 the signature (to determine that the signature was done by the key in question and that the message has
334 not changed) and then verification of the certificate chain (to determine that the key is bound to the right
335 identity).

336 Additionally, with certificates steps must be taken to ensure that the binding is currently valid—a
337 certificate typically has a “lifetime” built into it, but if a key is compromised during the life of the certificate
338 then the key-to-identity binding contained in the certificate becomes invalid while the certificate is still
339 valid on its face. Also, certificates often depend on associations that may end before their lifetime expires

340 (for example, certificates that should become invalid when someone changes employers, etc.) This
341 problem is solved by Certificate Revocation Lists (CRLs), which are lists of certificates from a given CA
342 that have been revoked since their issue. Another solution is the Online Certificate Status Protocol
343 (OCSP), which defines a method for calling servers to ask about the current validity of a given certificate.
344 Some of this same functionality is incorporated into the higher levels of the XML Key Management
345 Specification [XKMS], which allows requests to be made for “valid” keys.

346 A proper key management system is thus quite strong but very complex. Verifying a signature ends up
347 being a three-stage process of verifying the document-to-key binding, then verifying the key-to-identity
348 binding, then verifying the current validity of the key-to-document binding.

349 4.5 TLS/SSL Cipher Suites

350 The use of SSL 3.0 or TLS 1.0 [RFC2246] over HTTP is recommended at many places in this document.
351 However TLS/SSL can be configured to use many different cipher suites, not all of which are adequate to
352 provide “best practices” security. The following sections provide a brief description of cipher suites and
353 recommendations for cipher suite selection.

354 4.5.1 What Is a Cipher Suite?

355 **Note:** While references to the US Export restrictions are now obsolete, the constants
356 naming the cipher suites have not changed. Thus,
357 `SSL_DHE_DSS_EPORT_WITH_DES40_CBC_SHA` is still a valid cipher suite identifier,
358 and the explanation of the historical reasons for the inclusion of “EXPORT” has been left
359 in place in the following summary.

360 A cipher suite combines four kinds of security features, and is given a name in the SSL protocol
361 specification. Before data flows over a SSL connection, both ends attempt to negotiate a cipher suite.
362 This lets them establish an appropriate quality of protection for their communications, within the
363 constraints of the particular mechanism combinations which are available. The features associated with a
364 cipher suite are:

- 365 1. The type of key exchange algorithm used. SSL defines many; the ones that provide server
366 authentication are the most important ones, but anonymous key exchange is supported. (Note that
367 anonymous key exchange algorithms are subject to “man in the middle” attacks, and are **not**
368 **recommended** in the SAML context.) The “RSA” authenticated key exchange algorithm is currently
369 the most interoperable algorithm. Another important key exchange algorithm is the authenticated
370 Diffie-Hellman “DHE_DSS” key exchange, which has no patent-related implementation constraints.¹
- 371 2. Whether the key exchange algorithm is freely exportable from the United States of America.
372 Exportable algorithms must use short (512-bit) public keys for key exchange and short (40-bit)
373 symmetric keys for encryption. These keys are currently subject to breaking in an afternoon by a
374 moderately well-equipped adversary.
- 375 3. The encryption algorithm used. The fastest option is the RC4 stream cipher; DES and variants
376 (DES40, 3DES-EDE) are also supported in “cipher block chaining” (CBC) mode, as is null encryption
377 (in some suites). (Null encryption does nothing; in such cases SSL is used only to authenticate and
378 provide integrity protection. Cipher suites with null encryption do not provide confidentiality, and
379 **should not be used** in cases where confidentiality is a requirement.)
- 380 4. The digest algorithm used for the Message Authentication Code. The choices are MD5 and SHA1.

¹ The RSA patents have all expired; hence this issue is mostly historical.

381 For example, the cipher suite named SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA uses SSL,
382 uses an authenticated Diffie-Hellman key exchange (DHE_DSS), is export grade (EXPORT), uses an
383 exportable variant of the DES cipher (DES40_CBC), and uses the SHA1 digest algorithm in its MAC
384 (SHA).

385 A given implementation of SSL will support a particular set of cipher suites, and some subset of those will
386 be enabled by default. Applications have a limited degree of control over the cipher suites that are used
387 on their connections; they can enable or disable any of the supported cipher suites, but cannot change
388 the cipher suites that are available.

389 **4.5.2 Cipher Suite Recommendations**

390 The following cipher suites adequately meet SAML's requirements for confidentiality and message
391 integrity, and can be configured to meet the authentication requirement as well (by forcing the presence
392 of X.509v3 certificates). They are also well supported in many client applications. Support of these suites
393 is recommended:

- 394 • TLS_RSA_WITH_3DES_EDE_CBC_SHA (when using TLS)
- 395 • SSL_RSA_WITH_3DES_EDE_CBC_SHA (when using SSL)

396 However, the IETF is moving rapidly towards mandating the use of AES, which has both speed and
397 strength advantages. Forward-looking systems would be wise as well to implement support for the AES
398 cipher suites, such as:

- 399 • TLS_RSA_WITH_AES_128_CBC_SHA

400 **5 SAML-Specific Security Considerations**

401 The following sections analyze the security risks in using and implementing SAML and describe
402 countermeasures to mitigate the risks.

403 **5.1 SAML Assertions**

404 At the level of the SAML assertion itself, there is little to be said about security concerns—most concerns
405 arise during communications in the request/response protocol, or during the attempt to use SAML by
406 means of one of the bindings. However, one issue at the assertion level bears analysis: An assertion,
407 once issued, is out of the control of the issuer.

408 This fact has a number of ramifications. For example, the issuer has no control over how long the
409 assertion will be persisted in the systems of the consumer; nor does the issuer have control over the
410 parties with whom the consumer will share the assertion information. These concerns are over and above
411 concerns about a malicious attacker who can see the contents of assertions that pass over the wire
412 unencrypted (or insufficiently encrypted).

413 While efforts have been made to address many of these issues within the SAML specification, nothing
414 contained in the specification will erase the requirement for careful consideration of what to put in an
415 assertion. At all times, issuers should consider the possible consequences if the information in the
416 assertion is stored on a remote site, where it can be directly misused, or exposed to potential hackers, or
417 possibly stored for more creatively fraudulent uses. Issuers should also consider the possibility that the
418 information in the assertion could be shared with other parties, or even made public, either intentionally or
419 inadvertently.

420 **5.2 SAML Protocol**

421 The following sections describe security considerations for the SAML request-response protocol itself,
422 apart from any threats arising from use of a particular protocol binding.

423 **5.2.1 Denial of Service**

424 The SAML protocol is susceptible to a denial of service (DOS) attack. Handling a SAML request is
425 potentially a very expensive operation, including parsing the request message (typically involving
426 construction of a DOM tree), database/assertion store lookup (potentially on an unindexed key),
427 construction of a response message, and potentially one or more digital signature operations. Thus, the
428 effort required by an attacker generating requests is much lower than the effort needed to handle those
429 requests.

430 **5.2.1.1 Requiring Client Authentication at a Lower Level**

431 Requiring clients to authenticate at some level below the SAML protocol level (for example, using the
432 SOAP over HTTP binding, with HTTP over TLS/SSL, and with a requirement for client-side certificates
433 that have a trusted Certificate Authority at their root) will provide traceability in the case of a DOS attack.

434 If the authentication is used only to provide traceability, then this does not in itself prevent the attack from
435 occurring, but does function as a deterrent.

436 If the authentication is coupled with some access control system, then DOS attacks from non-insiders is
437 effectively blocked. (Note that it is possible that overloading the client-authentication scheme could still

438 function as a denial-of-service attack on the SAML service, but that this attack needs to be dealt with in
439 the context of the client authentication scheme chosen.)

440 Whatever system of client authentication is used, it should provide the ability to resolve a unique
441 originator for each request, and should not be subject to forgery. (For example, in the traceability-only
442 case, logging the IP address is insufficient since this information can easily be spoofed.)

443 **5.2.1.2 Requiring Signed Requests**

444 In addition to the benefits gained from client authentication discussed in Section 5.2.1.1, requiring a
445 signed request also lessens the order of the asymmetry between the work done by requester and
446 responder. The additional work required of the responder to verify the signature is a relatively small
447 percentage of the total work required of the responder, while the process of calculating the digital
448 signature represents a relatively large amount of work for the requester. Narrowing this asymmetry
449 decreases the risk associated with a DOS attack.

450 Note, however, that an attacker can theoretically capture a signed message and then replay it continually,
451 getting around this requirement. This situation can be avoided by requiring the use of the XML Signature
452 element `<ds:SignatureProperties>` containing a timestamp; the timestamp can then be used to
453 determine if the signature is recent. In this case, the narrower the window of time after issue that a
454 signature is treated as valid, the higher security you have against replay denial of service attacks.

455 **5.2.1.3 Restricting Access to the Interaction URL**

456 Limiting the ability to issue a request to a SAML service at a very low level to a set of known parties
457 drastically reduces the risk of a DOS attack. In this case, only attacks originating from within the finite set
458 of known parties are possible, greatly decreasing exposure both to potentially malicious clients and to
459 DOS attacks using compromised machines as zombies.

460 There are many possible methods of limiting access, such as placing the SAML responder inside a
461 secured intranet and implementing access rules at the router level.

462 **5.3 SAML Protocol Bindings**

463 The security considerations in the design of the SAML request-response protocol depend to a large
464 extent on the particular protocol binding (as defined in the SAML bindings specification **[SAMLBind]**) that
465 is used. Currently the only binding sanctioned by the OASIS Security Services Technical Committee is
466 the SOAP binding.

467 **5.3.1 SOAP Binding**

468 Since the SAML SOAP binding requires no authentication and has no requirements for either in-transit
469 confidentiality or message integrity, it is open to a wide variety of common attacks, which are detailed in
470 the following sections. General considerations are discussed separately from considerations related to
471 the SOAP-over-HTTP case.

472 **5.3.1.1 Eavesdropping**

473 Since there is no in-transit confidentiality requirement, it is possible that an eavesdropping party could
474 acquire both the SOAP message containing a request and the SOAP message containing the
475 corresponding response. This acquisition exposes both the nature of the request and the details of the
476 response, possibly including one or more assertions.

477 Exposure of the details of the request will in some cases weaken the security of the requesting party by
478 revealing details of what kinds of assertions it requires, or from whom those assertions are requested. For
479 example, if an eavesdropper can determine that site *X* is frequently requesting authentication assertions
480 with a given confirmation method from site *Y*, he may be able to use this information to aid in the
481 compromise of site *X*.

482 Similarly, eavesdropping on a series of authorization queries could create a “map” of resources that are
483 under the control of a given authorization authority.

484 Additionally, in some cases exposure of the request itself could constitute a violation of privacy. For
485 example, eavesdropping on a query and its response may expose that a given user is active on the
486 querying site, which could be information that should not be divulged in cases such as medical
487 information sites, political sites, and so on. Also the details of any assertions carried in the response may
488 be information that should be kept confidential. This is particularly true for responses containing attribute
489 assertions; if these attributes represent information that should not be available to entities not party to the
490 transaction (credit ratings, medical attributes, and so on), then the risk from eavesdropping is high.

491 In cases where any of these risks is a concern, the countermeasure for eavesdropping attacks is to
492 provide some form of in-transit message confidentiality. For SOAP messages, this confidentiality can be
493 enforced either at the SOAP level or at the SOAP transport level (or some level below it).

494 Adding in-transit confidentiality at the SOAP level means constructing the SOAP message such that,
495 regardless of SOAP transport, no one but the intended party will be able to access the message. The
496 general solution to this problem is likely to be XML Encryption [**XMLEnc**]. This specification allows
497 encryption of the SOAP message itself, which eliminates the risk of eavesdropping unless the key used in
498 the encryption has been compromised. Alternatively, deployers can depend on the SOAP transport layer,
499 or a layer beneath it, to provide in-transit confidentiality.

500 The details of how to provide this confidentiality depend on the specific SOAP transport chosen. Using
501 HTTP over TLS/SSL (described further in Section 5.3.2) is one method. Other transports will necessitate
502 other in-transit confidentiality techniques; for example, an SMTP transport might use S/MIME.

503 In some cases, a layer beneath the SOAP transport might provide the required in-transit confidentiality.
504 For example, if the request-response interaction is carried out over an IPsec tunnel, then adequate in-
505 transit confidentiality may be provided by the tunnel itself.

506 **5.3.1.2 Replay**

507 There is little vulnerability to replay attacks at the level of the SOAP binding. Replay is more of an issue in
508 the various profiles. The primary concern about replay at the SOAP binding level is the potential for use of
509 replay as a denial-of-service attack method.

510 In general, the best way to prevent replay attacks is to prevent the message capture in the first place.
511 Some of the transport-level schemes used to provide in-transit confidentiality will accomplish this goal.
512 For example, if the SAML request-response conversation occurs over SOAP on HTTP/TLS, third parties
513 are prevented from capturing the messages.

514 Note that since the potential replayer does not need to understand the message to replay it, schemes
515 such as XML Encryption do not provide protection against replay. If an attacker can capture a SAML
516 request that has been signed by the requester and encrypted to the responder, then the attacker can
517 replay that request at any time without needing to be able to undo the encryption. The SAML request
518 includes information about the issue time of the request, allowing a determination about whether replay is
519 occurring. Alternatively, the unique key of the request (its `RequestID`) can be used to determine if this is
520 a replay request or not.

521 Additional threats from the replay attack include cases where a “charge per request” model is in place.
522 Replay could be used to run up large charges on a given account.

523 Similarly, models where a client is allocated (or purchases) a fixed number of interactions with a system,
524 the replay attack could exhaust these uses unless the issuer is careful to keep track of the unique key of
525 each request.

526 **5.3.1.3 Message Insertion**

527 The message insertion attack for the SOAP binding amounts to the creation of a request. The ability to
528 make a request is not a threat at the SOAP binding level.

529 **5.3.1.4 Message Deletion**

530 The message deletion attack would either prevent a request from reaching a responder, or would prevent
531 the response from reaching the requester.

532 In either case, the SOAP binding does not address this threat. The SOAP protocol itself, and the
533 transports beneath it, may provide some information depending on how the message deletion is
534 accomplished.

535 Examples of reliable messaging systems that attenuate this risk include reliable HTTP (HTTPR) [**HTTPR**]
536 at the transport layer and the use of reliable messaging extensions in SOAP such as Microsoft's SRMP
537 for MSMQ [**SRMPPres**].

538 **5.3.1.5 Message Modification**

539 Message modification is a threat to the SOAP binding in both directions.

540 Modification of the request to alter the details of the request can result in significantly different results
541 being returned, which in turn can be used by a clever attacker to compromise systems depending on the
542 assertions returned. For example, altering the list of requested attributes in the
543 <AttributeDesignator> elements could produce results leading to compromise or rejection of the
544 request by the responder.

545 Modification of the request to alter the apparent issuer of the request could result in denial of service or
546 incorrect routing of the response. This alteration would need to occur below the SAML level and is thus
547 out of scope.

548 Modification of the response to alter the details of the assertions therein could result in vast degrees of
549 compromise. The simple examples of altering details of an authentication or an authorization decision
550 could lead to very serious security breaches.

551 In order to address these potential threats, a system that guarantees in-transit message integrity must be
552 used. The SAML protocol and the SOAP binding neither require nor forbid the deployment of systems that
553 guarantee in-transit message integrity, but due to this large threat, it is **highly recommended** that such a
554 system be used. At the SOAP binding level, this can be accomplished by digitally signing requests and
555 responses with a system such as XML Signature [**XMLSig**]. The SAML specification allows for such
556 signatures; see the SAML assertion and protocol specification [**SAMLCore**] for further information.

557 If messages are digitally signed (with a sensible key management infrastructure, see Section 4.4) then
558 the recipient has a guarantee that the message has not been altered in transit, unless the key used has
559 been compromised.

560 The goal of in-transit message integrity can also be accomplished at a lower level by using a SOAP
561 transport that provides the property of guaranteed integrity, or is based on a protocol that provides such a
562 property. SOAP over HTTP over TLS/SSL is a transport that would provide such a guarantee.

563 Encryption alone does not provide this protection, as even if the intercepted message could not be altered
564 per se, it could be replaced with a newly created one.

565 **5.3.1.6 Man-in-the-Middle**

566 The SOAP binding is susceptible to man-in-the-middle (MITM) attacks. In order to prevent malicious
567 entities from operating as a man in the middle (with all the perils discussed in both the eavesdropping and
568 message modification sections), some sort of bilateral authentication is required.

569 A bilateral authentication system would allow both parties to determine that what they are seeing in a
570 conversation actually came from the other party to the conversation.

571 At the SOAP binding level, this goal could also be accomplished by digitally signing both requests and
572 responses (with all the caveats discussed in Section 5.3.1.5 above). This method does not prevent an
573 eavesdropper from sitting in the middle and forwarding both ways, but he is prevented from altering the
574 conversation in any way without being detected.

575 Since many applications of SOAP do not use sessions, this sort of authentication of author (as opposed
576 to authentication of sender) may need to be combined with information from the transport layer to confirm
577 that the sender and the author are the same party in order to prevent a weaker form of "MITM as
578 eavesdropper".

579 Another implementation would depend on a SOAP transport that provides, or is implemented on a lower
580 layer that provides, bilateral authentication. The example of this is again SOAP over HTTP over TLS/SSL
581 with both server- and client-side certificates required.

582 Additionally, the validity interval of the assertions returned functions as an adjustment on the degree of
583 risk from MITM attacks. The shorter the valid window of the assertion, the less damage can be done if it is
584 intercepted.

585 **5.3.2 Specifics of SOAP over HTTP**

586 Since the SOAP binding requires that conformant applications support HTTP over TLS/SSL with a
587 number of different bilateral authentication methods such as Basic over server-side SSL and certificate-
588 backed authentication over server-side SSL, these methods are always available to mitigate threats in
589 cases where other lower-level systems are not available and the above listed attacks are considered
590 significant threats.

591 This does not mean that use of HTTP over TLS with some form of bilateral authentication is mandatory. If
592 an acceptable level of protection from the various risks can be arrived at through other means (for
593 example, by an IPsec tunnel), full TLS with certificates is not required. However, in the majority of cases
594 for SOAP over HTTP, using HTTP over TLS with bilateral authentication will be the appropriate choice.

595 Note, however, that the use of transport-level security (such as the SSL or TLS protocols under HTTP)
596 only provides confidentiality and/or integrity and/or authentication for "one hop". For models where there
597 may be intermediaries, or the assertions in question need to live over more than one hop, the use of
598 HTTP with TLS/SSL does not provide adequate security.

599 **5.4 Profiles of SAML**

600 The SAML bindings specification [**SAMLEndpoint**] in addition defines profiles of SAML, which are sets of
601 rules describing how to embed SAML assertions into and extract them from a framework or protocol.
602 Currently there are two profiles for SAML that are sanctioned by the OASIS Security Services Technical
603 Committee:

- 604 • Two web browser-based profiles that support single sign-on (SSO):
 - 605 – The browser/artifact profile for SAML
 - 606 – The browser/POST profile for SAML

607 (The OASIS Web Services Security Technical Committee has produced another profile of SAML, a draft
608 "SAML token profile" of the WSS specification [**WSS-SAML**] that describes how to use SAML assertions
609 to secure a web service message.)

610 **5.4.1 Web Browser-Based Profiles**

611 The following sections describe security considerations that are common to the browser/artifact and
612 browser/POST profiles for SAML.

613 Note that user authentication at the source site is explicitly out of scope, as are all issues that arise from
614 it. The key notion is that the source system entity must be able to ascertain that the authenticated client
615 system entity that it is interacting with is the same as the one in the next interaction step. One way to
616 accomplish this is for these initial steps to be performed using TLS as a session layer underneath the
617 protocol being used for this initial interaction (likely HTTP).

618 **5.4.1.1 Eavesdropping**

619 The possibility of eavesdropping exists in all web browser cases. In cases where confidentiality is
620 required (bearing in mind that any assertion that is not sent securely, along with the requests associated
621 with it, is available to the malicious eavesdropper), HTTP traffic needs to take place over a transport that
622 ensures confidentiality. HTTP over TLS/SSL [**RFC2246**] and the IP Security Protocol [**IPsec**] meet this
623 requirement.

624 The following sections provide more detail on the eavesdropping threat.

625 **5.4.1.1.1 Theft of the User Authentication Information**

626 In the case where the subject authenticates to the source site by revealing authentication information, for
627 example, in the form of a password, theft of the authentication information will enable an adversary to
628 impersonate the subject.

629 In order to avoid this problem, the connection between the subject's browser and the source site must
630 implement a confidentiality safeguard. In addition, steps must be taken by either the subject or the
631 destination site to ensure that the source site is genuinely the expected and trusted source site before
632 revealing the authentication information. Using HTTP over TLS can be used to address this concern.

633 **5.4.1.1.2 Theft of the Bearer Token**

634 In the case where the authentication assertion contains the assertion bearer's authentication protocol
635 identifier, theft of the artifact will enable an adversary to impersonate the subject.

636 Each of the following methods decreases the likelihood of this happening:

- 637 • The destination site implements a confidentiality safeguard on its connection with the subject's
638 browser.
- 639 • The subject or destination site ensures (out of band) that the source site implements a confidentiality
640 safeguard on its connection with the subject's browser.
- 641 • The destination site verifies that the subject's browser was directly redirected by a source site that
642 directly authenticated the subject.
- 643 • The source site refuses to respond to more than one request for an assertion corresponding to the
644 same assertion ID.

- 645 • If the assertion contains a condition element of type **AudienceRestrictionConditionType** that
646 identifies a specific domain, then the destination site verifies that it is a member of that domain.
- 647 • The connection between the destination site and the source site, over which the assertion ID is
648 passed, is implemented with a confidentiality safeguard.
- 649 • The destination site, in its communication with the source site, over which the assertion ID is passed,
650 must verify that the source site is genuinely the expected and trusted source site.

651 **5.4.1.2 Replay**

652 The possibility of a replay attack exists for this set of profiles. A replay attack can be used either to
653 attempt to deny service or to retrieve information fraudulently. The specific countermeasures depend on
654 which specific profile is being used, and thus are discussed in Sections 5.4.2.1 and 5.4.3.1.

655 **5.4.1.3 Message Insertion**

656 Message insertion attacks are not a general threat in this set of profiles.

657 **5.4.1.4 Message Deletion**

658 Deleting a message during any step of the interactions between the browser, SAML assertion issuer, and
659 SAML assertion consumer will cause the interaction to fail. It results in a denial of some service but does
660 not increase the exposure of any information.

661 The SAML bindings and profiles specification provides no countermeasures for message deletion.

662 **5.4.1.5 Message Modification**

663 The possibility of alteration of the messages in the stream exists for this set of profiles. Some potential
664 undesirable results are as follows:

- 665 • Alteration of the initial request can result in rejection at the SAML issuer, or creation of an artifact
666 targeted at a different resource than the one requested
- 667 • Alteration of the artifact can result in denial of service at the SAML consumer.
- 668 • Alteration of the assertions themselves while in transit could result in all kinds of bad results (if they
669 are unsigned) or denial of service (if they are signed and the consumer rejects them).

670 To avoid message modification, the traffic needs to be transported by means of a system that guarantees
671 message integrity from endpoint to endpoint.

672 For the web browser-based profiles, the recommended method of providing message integrity in transit is
673 the use of HTTP over TLS/SSL with a cipher suite that provides data integrity checking.

674 **5.4.1.6 Man-in-the-Middle**

675 Man-in-the-middle attacks are particularly pernicious for this set of profiles. The MITM can relay requests,
676 capture the returned assertion (or artifact), and relay back a false one. Then the original user cannot
677 access the resource in question, but the MITM can do so using the captured resource.

678 Preventing this threat requires a number of countermeasures. First, using a system that provides strong
679 bilateral authentication will make it much more difficult for a MITM to insert himself into the conversation.

680 However the possibility still exists of a MITM who is purely acting as a bidirectional port forwarder, and
681 eavesdropping on the information with the intent to capture the returned assertion or handler (and
682 possibly alter the final return to the requester). Putting a confidentiality system in place will prevent
683 eavesdropping. Putting a data integrity system in place will prevent alteration of the message during port
684 forwarding.

685 For this set of profiles, all the requirements of strong bilateral session authentication, confidentiality, and
686 data integrity can be met by the use of HTTP over TLS/SSL if the TLS/SSL layer uses an appropriate
687 cipher suite (strong enough encryption to provide confidentiality, and supporting data integrity) and
688 requires X509v3 certificates for authentication.

689 **5.4.2 Browser/Artifact Profile**

690 Many specific threats and counter-measures for the Browser/Artifact profile are documented normatively
691 in the SAML bindings specification **[SAMLBind]**. Additional non-normative comments are included below.

692 **5.4.2.1 Replay**

693 The threat of replay as a reuse of an artifact is addressed by the requirement that each artifact is a one-
694 time-use item. Systems should track cases where multiple requests are made referencing the same
695 artifact, as this situation may represent intrusion attempts.

696 The threat of replay on the original request that results in the assertion generation is not addressed by
697 SAML, but should be mitigated by the original authentication process.

698 **5.4.3 Browser/POST Profile**

699 Many specific threats and counter-measures for the Browser/POST profile are documented normatively in
700 the SAML bindings specification **[SAMLBind]**. Additional non-normative comments are included below.

701 **5.4.3.1 Replay**

702 Replay attacks amount to resubmission of the form in order to access a protected resource fraudulently.
703 The profile mandates that the assertions transferred have the one-use property at the destination site,
704 preventing replay attacks from succeeding.

705

6 References

706 The following are cited in the text of this document:

- 707 **[Anonymity]** Anonymity, Unobservability, and Pseudonymity -- A Proposal for Terminology
708 Andreas Pfitzmann, Marit Köhntopp,
709 http://www.realname-diskussion.info/anon_terminology.pdf.
- 710 **[FreeHaven]** The Free Haven Project: Distributed Anonymous Storage Service
711 Roger Dingledine & Michael J. Freedman & David Molnar
712 <http://www.freehaven.net/paper/node6.html>
713 <http://www.freehaven.net/paper/node7.html>
- 714 **[HTTPR]** A Primer for HTTPR: An overview of the reliable HTTP protocol
715 Stephen Todd, Francis Parr, Michael H. Conner
716 <http://www-106.ibm.com/developerworks/webservices/library/ws-phtt/>
- 717 **[IPsec]** IETF IP Security Protocol Working Group, [http://www.ietf.org/html.charters/ipsec-](http://www.ietf.org/html.charters/ipsec-charter.html)
718 [charter.html](http://www.ietf.org/html.charters/ipsec-charter.html).
- 719 **[Pooling]** Pooling Intellectual Capital: Thoughts on Anonymity, Pseudonymity, and Limited
720 Liability in Cyberspace
721 David G. Post
722 <http://www.cli.org/DPost/paper8.htm>
- 723 **[Rescorla-Sec]** E. Rescorla et al., *Guidelines for Writing RFC Text on Security Considerations*,
724 <http://www.ietf.org/internet-drafts/draft-rescorla-sec-cons-03.txt>.
- 725 **[RFC2246]** The TLS Protocol Version 1.0, <http://www.ietf.org/rfc/rfc2246.html>.
- 726 **[SAMLBind]** Prateek Mishra et al., *Bindings and Profiles for the OASIS Security Assertion*
727 *Markup Language (SAML)*, <http://www.oasis-open.org/committees/security/>,
728 OASIS, May 2003.
- 729 **[SAMLCore]** Phillip Hallam-Baker et al., *Assertions and Protocol for the OASIS Security*
730 *Assertion Markup Language (SAML)*, [http://www.oasis-](http://www.oasis-open.org/committees/security/)
731 [open.org/committees/security/](http://www.oasis-open.org/committees/security/), OASIS, May 2003.
- 732 **[SAMLGloss]** Jeff Hodges et al., *Glossary for the OASIS Security Assertion Markup Language*
733 *(SAML)*, <http://www.oasis-open.org/committees/security/>, OASIS, May 2003.
- 734 **[SRMPPres]** Message Queuing: Messaging Over The Internet
735 Shai Kariv
736 <http://www.microsoft.com/israel/events/teched/presentations/EN308.zip>
- 737 **[WSS]** Web Services Security specifications (WSS), OASIS. [http://www.oasis-](http://www.oasis-open.org/committees/wss)
738 [open.org/committees/wss](http://www.oasis-open.org/committees/wss).
- 739 **[WSS-SAML]** P. Hallam-Baker et al., *Web Services Security: SAML Token Profile*, OASIS,
740 March 2003, <http://www.oasis-open.org/committees/wss>.
- 741 **[XKMS]** XML Key Management Specifications, W3C. <http://www.w3.org/2001/XKMS/>.
- 742 **[XMLEnc]** Donald Eastlake et al., *XML Encryption Syntax and Processing*,
743 <http://www.w3.org/TR/xmlenc-core/>, World Wide Web Consortium, December
744 2002.
- 745 **[XMLSig]** Donald Eastlake et al., *XML-Signature Syntax and Processing*,
746 <http://www.w3.org/TR/xmlsig-core/>, World Wide Web Consortium.

747 The following additional documents are recommended reading:

- 748 **[ebXML-MSS]** Message Service Specification V2.0, OASIS, April 2002. [http://www.oasis-](http://www.oasis-open.org/committees/download.php/272/ebMS_v2_0.pdf)
749 [open.org/committees/download.php/272/ebMS_v2_0.pdf](http://www.oasis-open.org/committees/download.php/272/ebMS_v2_0.pdf). The information about
750 the security module is the material of interest.
- 751 **[ebXML-Risk]** ebXML Technical Architecture Risk Assessment v1.0,
752 <http://www.ebxml.org/specs/secRISK.pdf>.
- 753 **[Prudent]** Prudent Engineering Practice for Cryptographic Protocols,
754 <http://citeseer.nj.nec.com/abadi96prudent.html>.
- 755 **[Robustness]** Robustness principles for public key protocols,
756 <http://citeseer.nj.nec.com/2927.html>.

757 **Appendix A. Acknowledgments**

758 The editors would like to acknowledge the contributions of the OASIS SAML Technical Committee, whose
759 voting members at the time of publication were:

- 760 • Irving Reid, Baltimore Technologies
- 761 • Hal Lockhart, BEA Systems
- 762 • Ronald Jacobson, Computer Associates
- 763 • John Hughes, Entegriy Solutions
- 764 • Carlisle Adams, Entrust
- 765 • Robert Griffin, Entrust
- 766 • Scott Cantor, Individual
- 767 • Bob Morgan, Individual
- 768 • Clifford Thompson, Individual
- 769 • Padraig Moloney, NASA
- 770 • Prateek Mishra, Netegrity (co-chair)
- 771 • Frederick Hirsch, Nokia
- 772 • Senthil Sengodan, Nokia
- 773 • Timo Skytta, Nokia
- 774 • Charles Knouse, Oblix
- 775 • Steve Anderson, OpenNetwork
- 776 • Simon Godik, OverXeer
- 777 • Rob Philpott, RSA Security (co-chair)
- 778 • Dipak Chopra, SAP
- 779 • Jahan Moreh, Sigaba
- 780 • Bhavna Bhatnagar, Sun Microsystems
- 781 • Jeff Hodges, Sun Microsystems
- 782 • Eve Maler, Sun Microsystems (coordinating editor)
- 783 • Emily Xu, Sun Microsystems
- 784 • Phillip Hallam-Baker, VeriSign

785

Appendix B. Notices

786 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
787 might be claimed to pertain to the implementation or use of the technology described in this document or
788 the extent to which any license under such rights might or might not be available; neither does it
789 represent that it has made any effort to identify any such rights. Information on OASIS's procedures with
790 respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights
791 made available for publication and any assurances of licenses to be made available, or the result of an
792 attempt made to obtain a general license or permission for the use of such proprietary rights by
793 implementors or users of this specification, can be obtained from the OASIS Executive Director.

794 OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications,
795 or other proprietary rights which may cover technology that may be required to implement this
796 specification. Please address the information to the OASIS Executive Director.

797 **Copyright © OASIS Open 2003. All Rights Reserved.**

798 This document and translations of it may be copied and furnished to others, and derivative works that
799 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published
800 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice
801 and this paragraph are included on all such copies and derivative works. However, this document itself
802 may not be modified in any way, such as by removing the copyright notice or references to OASIS,
803 except as needed for the purpose of developing OASIS specifications, in which case the procedures for
804 copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required to
805 translate it into languages other than English.

806 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
807 or assigns.

808 This document and the information contained herein is provided on an "AS IS" basis and OASIS
809 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
810 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR
811 ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

812

Appendix C. Revision History

Draft	Who	What
01	Eve Maler	Cosmetic changes to bring spec up to 1.1 WD status. Copyedits and changes to bring SAML technology references up to date. Also updated various external references.
02	Rob Philpott	Updated bibliography dates for all SAML specs. Accepted all changes in document for Last Call.

813