



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

Committee Specification, 18 July 2003

Document identifier:

sstc-saml-sec-consider-1.1-cs-02

Location:

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

Editor:

Eve Maler, Sun Microsystems (eve.maler@sun.com)
Rob Philpott, RSA Security (rphilpott@rsasecurity.com)

Contributors:

Hal Lockhart, BEA Systems, Inc.
Tim Moses, Entrust
Evan Prodromou, former member
Marlena Erdos, IBM
RL "Bob" Morgan, individual
Chris McLaren, Netegrity (former editor)
Prateek Mishra, Netegrity
Jeff Hodges, Sun Microsystems

Abstract:

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

Status:

This document is a **Committee Specification** of the OASIS Security Services Technical Committee. This document is updated periodically on no particular schedule. Send comments to the editors.

Committee members should send comments on this specification to the security-services@lists.oasis-open.org list. Others should subscribe to and send comments to the security-services-comment@lists.oasis-open.org list. 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/>).

37 For information on errata discovered in this specification, please refer to the most recent errata
38 document which can be found in the document repository at the Security Services TC web page
39 (<http://www.oasis-open.org/committees/security/>).

Table of Contents

41	1	Introduction.....	5
42	2	Privacy.....	6
43	2.1	Ensuring Confidentiality.....	6
44	2.2	Notes on Anonymity.....	6
45	2.2.1	Definitions That Relate to Anonymity.....	6
46	2.2.2	Pseudonymity and Anonymity.....	7
47	2.2.3	Behavior and Anonymity.....	8
48	2.2.4	Implications for Privacy.....	8
49	3	Security.....	9
50	3.1	Background.....	9
51	3.2	Scope.....	9
52	3.3	SAML Threat Model.....	9
53	4	Security Techniques.....	11
54	4.1	Authentication.....	11
55	4.1.1	Active Session.....	11
56	4.1.2	Message-Level.....	11
57	4.2	Confidentiality.....	11
58	4.2.1	In Transit.....	11
59	4.2.2	Message-Level.....	11
60	4.3	Data Integrity.....	11
61	4.3.1	In Transit.....	12
62	4.3.2	Message-Level.....	12
63	4.4	Notes on Key Management.....	12
64	4.4.1	Access to the Key.....	12
65	4.4.2	Binding of Identity to Key.....	12
66	4.5	TLS/SSL Cipher Suites.....	13
67	4.5.1	What Is a Cipher Suite?.....	13
68	4.5.2	Cipher Suite Recommendations.....	14
69	5	SAML-Specific Security Considerations.....	15
70	5.1	SAML Assertions.....	15
71	5.2	SAML Protocol.....	15
72	5.2.1	Denial of Service.....	15
73	5.2.1.1	Requiring Client Authentication at a Lower Level.....	15
74	5.2.1.2	Requiring Signed Requests.....	16

75	5.2.1.3 Restricting Access to the Interaction URL.....	16
76	5.3 SAML Protocol Bindings.....	16
77	5.3.1 SOAP Binding.....	16
78	5.3.1.1 Eavesdropping.....	16
79	5.3.1.2 Replay.....	17
80	5.3.1.3 Message Insertion.....	18
81	5.3.1.4 Message Deletion.....	18
82	5.3.1.5 Message Modification.....	18
83	5.3.1.6 Man-in-the-Middle.....	19
84	5.3.2 Specifics of SOAP over HTTP.....	19
85	5.4 Profiles of SAML.....	19
86	5.4.1 Web Browser-Based Profiles.....	20
87	5.4.1.1 Eavesdropping.....	20
88	5.4.1.1.1 Theft of the User Authentication Information.....	20
89	5.4.1.1.2 Theft of the Bearer Token.....	20
90	5.4.1.2 Replay.....	21
91	5.4.1.3 Message Insertion.....	21
92	5.4.1.4 Message Deletion.....	21
93	5.4.1.5 Message Modification.....	21
94	5.4.1.6 Man-in-the-Middle.....	21
95	5.4.2 Browser/Artifact Profile.....	22
96	5.4.2.1 Replay.....	22
97	5.4.3 Browser/POST Profile.....	22
98	5.4.3.1 Replay.....	22
99	6 References.....	23
100	Appendix A. Acknowledgments.....	25
101	Appendix B. Notices.....	26
102		

103

1 Introduction

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

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

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

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

117 2 Privacy

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

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

126 2.1 Ensuring Confidentiality

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

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

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

141 2.2 Notes on Anonymity

142 The following sections discuss the concept of anonymity.

143 2.2.1 Definitions That Relate to Anonymity

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

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

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

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

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

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

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

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

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

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

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

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

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

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

181 **2.2.2 Pseudonymity and Anonymity**

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

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

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

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

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

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

198 **2.2.3 Behavior and Anonymity**

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

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

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

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

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

212 **2.2.4 Implications for Privacy**

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

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

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

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

221 3 Security

222 The following sections discuss security considerations.

223 3.1 Background

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

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

236 3.2 Scope

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

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

253 3.3 SAML Threat Model

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

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

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

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

272 4 Security Techniques

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

275 4.1 Authentication

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

278 4.1.1 Active Session

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

284 4.1.2 Message-Level

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

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

291 4.2 Confidentiality

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

294 4.2.1 In Transit

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

297 4.2.2 Message-Level

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

300 4.3 Data Integrity

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

303 **4.3.1 In Transit**

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

307 **4.3.2 Message-Level**

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

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

312 **4.4 Notes on Key Management**

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

317 **4.4.1 Access to the Key**

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

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

326 **4.4.2 Binding of Identity to Key**

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

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

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

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

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

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

353 4.5 TLS/SSL Cipher Suites

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

358 4.5.1 What Is a Cipher Suite?

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

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

- 369 1. The type of key exchange algorithm used. SSL defines many; the ones that provide server
370 authentication are the most important ones, but anonymous key exchange is supported. (Note that
371 anonymous key exchange algorithms are subject to “man in the middle” attacks, and are **not**
372 **recommended** in the SAML context.) The “RSA” authenticated key exchange algorithm is currently
373 the most interoperable algorithm. Another important key exchange algorithm is the authenticated
374 Diffie-Hellman “DHE_DSS” key exchange, which has no patent-related implementation constraints.¹
- 375 2. Whether the key exchange algorithm is freely exportable from the United States of America.
376 Exportable algorithms must use short (512-bit) public keys for key exchange and short (40-bit)
377 symmetric keys for encryption. These keys are currently subject to breaking in an afternoon by a
378 moderately well-equipped adversary.
- 379 3. The encryption algorithm used. The fastest option is the RC4 stream cipher; DES and variants
380 (DES40, 3DES-EDE) are also supported in “cipher block chaining” (CBC) mode, as is null encryption
381 (in some suites). (Null encryption does nothing; in such cases SSL is used only to authenticate and
382 provide integrity protection. Cipher suites with null encryption do not provide confidentiality, and
383 **should not be used** in cases where confidentiality is a requirement.)
- 384 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.

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

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

393 **4.5.2 Cipher Suite Recommendations**

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

- 398 • TLS_RSA_WITH_3DES_EDE_CBC_SHA (when using TLS)
- 399 • SSL_RSA_WITH_3DES_EDE_CBC_SHA (when using SSL)

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

- 403 • TLS_RSA_WITH_AES_128_CBC_SHA

404 5 SAML-Specific Security Considerations

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

407 5.1 SAML Assertions

408 At the level of the SAML assertion itself, there is little to be said about security concerns—most concerns
409 arise during communications in the request/response protocol, or during the attempt to use SAML by
410 means of one of the bindings. The consumer is, of course, always expected to honor the validity interval
411 of the assertion and any `<DoNotCacheCondition>` elements that are present in the assertion.

412 However, one issue at the assertion level bears analysis: an assertion, once issued, is out of the control
413 of the issuer. This fact has a number of ramifications. For example, the issuer has no control over how
414 long the assertion will be persisted in the systems of the consumer; nor does the issuer have control over
415 the parties with whom the consumer will share the assertion information. These concerns are over and
416 above concerns about a malicious attacker who can see the contents of assertions that pass over the
417 wire unencrypted (or insufficiently encrypted).

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

425 5.2 SAML Protocol

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

428 5.2.1 Denial of Service

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

435 5.2.1.1 Requiring Client Authentication at a Lower Level

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

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

441 If the authentication is coupled with some access control system, then DOS attacks from non-insiders is
442 effectively blocked. (Note that it is possible that overloading the client-authentication scheme could still
443 function as a denial-of-service attack on the SAML service, but that this attack needs to be dealt with in
444 the context of the client authentication scheme chosen.)

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

448 **5.2.1.2 Requiring Signed Requests**

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

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

460 **5.2.1.3 Restricting Access to the Interaction URL**

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

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

467 **5.3 SAML Protocol Bindings**

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

472 **5.3.1 SOAP Binding**

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

477 **5.3.1.1 Eavesdropping**

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

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

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

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

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

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

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

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

511 **5.3.1.2 Replay**

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

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

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

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

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

531 **5.3.1.3 Message Insertion**

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

534 **5.3.1.4 Message Deletion**

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

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

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

543 **5.3.1.5 Message Modification**

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

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

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

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

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

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

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

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

570 **5.3.1.6 Man-in-the-Middle**

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

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

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

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

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

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

590 **5.3.2 Specifics of SOAP over HTTP**

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

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

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

604 **5.4 Profiles of SAML**

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

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

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

615 **5.4.1 Web Browser-Based Profiles**

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

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

623 **5.4.1.1 Eavesdropping**

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

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

630 **5.4.1.1.1 Theft of the User Authentication Information**

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

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

638 **5.4.1.1.2 Theft of the Bearer Token**

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

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

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

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

656 **5.4.1.2 Replay**

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

660 **5.4.1.3 Message Insertion**

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

662 **5.4.1.4 Message Deletion**

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

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

667 **5.4.1.5 Message Modification**

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

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

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

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

679 **5.4.1.6 Man-in-the-Middle**

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

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

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

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

694 **5.4.2 Browser/Artifact Profile**

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

697 **5.4.2.1 Replay**

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

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

703 **5.4.3 Browser/POST Profile**

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

706 **5.4.3.1 Replay**

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

710

6 References

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

- 712 **[Anonymity]** Anonymity, Unobservability, and Pseudonymity -- A Proposal for Terminology
713 Andreas Pfitzmann, Marit Köhntopp,
714 http://www.realname-diskussion.info/anon_terminology.pdf.
- 715 **[FreeHaven]** The Free Haven Project: Distributed Anonymous Storage Service
716 Roger Dingledine & Michael J. Freedman & David Molnar
717 <http://www.freehaven.net/paper/node6.html>
718 <http://www.freehaven.net/paper/node7.html>
- 719 **[HTTPR]** A Primer for HTTPR: An overview of the reliable HTTP protocol
720 Stephen Todd, Francis Parr, Michael H. Conner
721 <http://www-106.ibm.com/developerworks/webservices/library/ws-phtt/>
- 722 **[IPsec]** IETF IP Security Protocol Working Group, [http://www.ietf.org/html.charters/ipsec-](http://www.ietf.org/html.charters/ipsec-charter.html)
723 [charter.html](http://www.ietf.org/html.charters/ipsec-charter.html).
- 724 **[Pooling]** Pooling Intellectual Capital: Thoughts on Anonymity, Pseudonymity, and Limited
725 Liability in Cyberspace
726 David G. Post
727 <http://www.cli.org/DPost/paper8.htm>
- 728 **[Rescorla-Sec]** E. Rescorla et al., *Guidelines for Writing RFC Text on Security Considerations*,
729 <http://www.ietf.org/internet-drafts/draft-rescorla-sec-cons-03.txt>.
- 730 **[RFC2246]** The TLS Protocol Version 1.0, <http://www.ietf.org/rfc/rfc2246.html>.
- 731 **[SAMLBind]** E. Maler et al., *Bindings and Profiles for the OASIS Security Assertion Markup*
732 *Language (SAML)*, <http://www.oasis-open.org/committees/security/>, OASIS, May
733 2003.
- 734 **[SAMLCore]** E. Maler et al., *Assertions and Protocol for the OASIS Security Assertion Markup*
735 *Language (SAML)*, <http://www.oasis-open.org/committees/security/>, OASIS, May
736 2003.
- 737 **[SAMLGloss]** E. Maler et al., *Glossary for the OASIS Security Assertion Markup Language*
738 *(SAML)*, <http://www.oasis-open.org/committees/security/>, OASIS, May 2003.
- 739 **[SRMPPres]** Message Queuing: Messaging Over The Internet
740 Shai Kariv
741 <http://www.microsoft.com/israel/events/teched/presentations/EN308.zip>
- 742 **[WSS]** Web Services Security specifications (WSS), OASIS. [http://www.oasis-](http://www.oasis-open.org/committees/wss)
743 [open.org/committees/wss](http://www.oasis-open.org/committees/wss).
- 744 **[WSS-SAML]** P. Hallam-Baker et al., *Web Services Security: SAML Token Profile*, OASIS,
745 March 2003, <http://www.oasis-open.org/committees/wss>.
- 746 **[XKMS]** XML Key Management Specifications, W3C. <http://www.w3.org/2001/XKMS/>.
- 747 **[XMLEnc]** Donald Eastlake et al., *XML Encryption Syntax and Processing*,
748 <http://www.w3.org/TR/xmlenc-core/>, World Wide Web Consortium, December
749 2002.
- 750 **[XMLSig]** Donald Eastlake et al., *XML-Signature Syntax and Processing*,
751 <http://www.w3.org/TR/xmlsig-core/>, World Wide Web Consortium.

752 The following additional documents are recommended reading:

753 **[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)
754 [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
755 the security module is the material of interest.

756 **[ebXML-Risk]** ebXML Technical Architecture Risk Assessment v1.0,
757 <http://www.ebxml.org/specs/secRISK.pdf>.

758 **[Prudent]** Prudent Engineering Practice for Cryptographic Protocols,
759 <http://citeseer.nj.nec.com/abadi96prudent.html>.

760 **[Robustness]** Robustness principles for public key protocols,
761 <http://citeseer.nj.nec.com/2927.html>.

762 **Appendix A. Acknowledgments**

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

- 765 • Frank Siebenlist, Argonne National Laboratory
- 766 • Irving Reid, Baltimore Technologies
- 767 • Hal Lockhart, BEA Systems
- 768 • Steven Lewis, Booz Allen Hamilton
- 769 • John Hughes, Entegriety Solutions
- 770 • Carlisle Adams, Entrust
- 771 • Jason Rouault, Hewlett-Packard
- 772 • Maryann Hondo, IBM
- 773 • Anthony Nadalin, IBM
- 774 • Scott Cantor, individual
- 775 • RL “Bob” Morgan, individual
- 776 • Trevor Perrin, individual
- 777 • Padraig Moloney, NASA
- 778 • Prateek Mishra, Netegrity (co-chair)
- 779 • Frederick Hirsch, Nokia
- 780 • Senthil Sengodan, Nokia
- 781 • Timo Skytta, Nokia
- 782 • Charles Knouse, Oblix
- 783 • Steve Anderson, OpenNetwork
- 784 • Simon Godik, Overxeer
- 785 • Rob Philpott, RSA Security (co-chair)
- 786 • Dipak Chopra, SAP
- 787 • Jahan Moreh, Sigaba
- 788 • Bhavna Bhatnagar, Sun Microsystems
- 789 • Jeff Hodges, Sun Microsystems
- 790 • Eve Maler, Sun Microsystems (coordinating editor)
- 791 • Emily Xu, Sun Microsystems
- 792 • Phillip Hallam-Baker, VeriSign

793

Appendix B. Notices

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

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

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

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

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

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