



Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0

Committee Draft 04, 15 January 2005

Document identifier:

sstc-saml-sec-consider-2.0-cd-04

Location:

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

Editors:

Frederick Hirsch, Nokia
Rob Philpott, RSA Security
Eve Maler, Sun Microsystems

SAML V2.0 Contributors:

Conor P. Cahill, AOL
John Hughes, Atos Origin
Hal Lockhart, BEA Systems
Michael Beach, Boeing
Rebekah Metz, Booz Allen Hamilton
Rick Randall, Booz, Allen, Hamilton
Tim Alsop, CyberSafe Limited
Thomas Wisniewski, Entrust
Irving Reid, Hewlett-Packard
Paula Austel, IBM
Maryann Hondo, IBM
Michael McIntosh, IBM
Tony Nadalin, IBM
Nick Ragouzis, Individual
Scott Cantor, Internet2
RL 'Bob' Morgan, Internet2
Peter C Davis, Neustar
Jeff Hodges, Neustar
Frederick Hirsch, Nokia
John Kemp, Nokia
Paul Madsen, NTT
Charles Knouse, Oblix
Steve Anderson, OpenNetwork
Prateek Mishra, Principal Identity
John Linn, RSA Security
Rob Philpott, RSA Security
Jahan Moreh, Sigaba
Anne Anderson, Sun Microsystems
Gary Ellison, Sun Microsystems
Eve Maler, Sun Microsystems
Ron Monzillo, Sun Microsystems

45 Greg Whitehead, Trustgenix

46 **Abstract:**

47 This non-normative specification describes and analyzes the security and privacy properties of
48 SAML.

49 **Status:**

50 This is a **Committee Draft** approved by the Security Services Technical Committee on 15
51 January 2005.

52 Committee members should submit comments and potential errata to the [security-](mailto:security-services@lists.oasis-open.org)
53 [services@lists.oasis-open.org](mailto:security-services@lists.oasis-open.org) list. Others should submit them by filling out the web form located
54 at http://www.oasis-open.org/committees/comments/form.php?wg_abbrev=security. The
55 committee will publish on its web page (<http://www.oasis-open.org/committees/security>) a catalog
56 of any changes made to this document.

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

61 Table of Contents

62	1 Introduction.....	5
63	2 Privacy.....	6
64	2.1 Ensuring Confidentiality.....	6
65	2.2 Notes on Anonymity.....	6
66	2.2.1 Definitions That Relate to Anonymity	6
67	2.2.2 Pseudonymity and Anonymity.....	7
68	2.2.3 Behavior and Anonymity.....	7
69	2.2.4 Implications for Privacy.....	8
70	3 Security.....	9
71	3.1 Background.....	9
72	3.2 Scope.....	9
73	3.3 SAML Threat Model.....	9
74	4 Security Techniques.....	11
75	4.1 Authentication.....	11
76	4.1.1 Active Session.....	11
77	4.1.2 Message-Level.....	11
78	4.2 Confidentiality.....	11
79	4.2.1 In Transit.....	11
80	4.2.2 Message-Level.....	11
81	4.3 Data Integrity.....	11
82	4.3.1 In Transit.....	11
83	4.3.2 Message-Level.....	11
84	4.4 Notes on Key Management.....	12
85	4.4.1 Access to the Key.....	12
86	4.4.2 Binding of Identity to Key.....	12
87	4.5 SSL/TLS Cipher Suites.....	12
88	4.5.1 SSL/TLS Cipher Suites.....	13
89	4.5.2 SSL/TLS Recommendations.....	14
90	5 General SAML Security Considerations.....	15
91	5.1 SAML Assertions.....	15
92	5.2 SAML Protocol.....	15
93	5.2.1 Denial of Service.....	15
94	5.2.1.1 Requiring Client Authentication at a Lower Level.....	15
95	5.2.1.2 Requiring Signed Requests.....	16
96	5.2.1.3 Restricting Access to the Interaction URL.....	16
97	6 SAML Bindings Security Considerations.....	17
98	6.1 SAML SOAP Binding.....	17
99	6.1.1 Eavesdropping.....	17
100	6.1.2 Replay.....	18
101	6.1.3 Message Insertion.....	18
102	6.1.4 Message Deletion.....	18
103	6.1.5 Message Modification.....	18
104	6.1.6 Man-in-the-Middle.....	19
105	6.1.7 Use of SOAP over HTTP.....	19

106	6.2 Reverse SOAP (PAOS) Binding.....	20
107	6.2.1 Denial of Service.....	20
108	6.3 HTTP Redirect binding.....	20
109	6.3.1 Denial of Service.....	20
110	6.4 HTTP Redirect/POST binding.....	20
111	6.4.1 Stolen Assertion.....	20
112	6.4.2 Man In the Middle Attack.....	21
113	6.4.3 Forged Assertion.....	21
114	6.4.4 Browser State Exposure.....	21
115	6.4.5 Replay.....	21
116	6.4.6 Modification or Exposure of state information.....	21
117	6.5 HTTP Artifact Binding.....	22
118	6.5.1 Stolen Artifact	22
119	6.5.2 Attacks on the SAML Protocol Message Exchange.....	22
120	6.5.3 Malicious Destination Site.....	22
121	6.5.4 Forged SAML Artifact.....	23
122	6.5.5 Browser State Exposure.....	23
123	6.5.6 Replay.....	23
124	6.6 SAML URI Binding.....	23
125	6.6.1 Substitution.....	23
126	7 SAML Profile Security Considerations.....	24
127	7.1 Web Browser Single Sign-On (SSO) Profiles.....	24
128	7.1.1 SSO Profile.....	24
129	7.1.1.1 Eavesdropping.....	24
130	7.1.1.2 Theft of the User Authentication Information.....	24
131	7.1.1.3 Theft of the Bearer Token.....	24
132	7.1.1.4 Replay.....	25
133	7.1.1.5 Message Insertion.....	25
134	7.1.1.6 Message Deletion.....	25
135	7.1.1.7 Message Modification.....	25
136	7.1.1.8 Man-in-the-Middle.....	25
137	7.1.1.9 Impersonation without Reauthentication.....	26
138	7.1.2 Enhanced Client and Proxy Profile.....	26
139	7.1.2.1 Man in the Middle.....	26
140	7.1.2.2 Denial of Service.....	26
141	7.1.3 Identity Provider Discovery Profile.....	26
142	7.1.4 Single Logout Profile.....	26
143	7.2 Name Identifier Management Profiles.....	27
144	7.3 Attribute Profiles.....	27
145	8 Summary.....	28
146	9 References.....	29

147

1 Introduction

148 This non-normative document describes and analyzes the security and privacy properties of the OASIS
149 Security Assertion Markup Language (SAML) defined in the core SAML specification [SAMLCore] and the
150 SAML bindings [SAMLBind] and profiles [SAMLProf] specifications. The intent in this document is to
151 provide information to architects, implementors, and reviewers of SAML-based systems about the
152 following:

- 153 • The privacy issues to be considered and how SAML architecture addresses these issues
- 154 • The threats, and thus security risks, to which a SAML-based system is subject
- 155 • The security risks the SAML architecture addresses, and how it does so
- 156 • The security risks it does not address
- 157 • Recommendations for countermeasures that mitigate those security risks

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

159 The rest of this section describes the background and assumptions underlying the analysis in this
160 document. Section 4 provides a high-level view of security techniques and technologies that should be
161 used with SAML. The following sections analyze the risks associated with the SAML assertions and
162 protocol as well as specific risks associated with SAML bindings and profiles.

2 Privacy

163

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

169 Many countries and jurisdictions have laws and regulations regarding privacy and these should be
170 considered when deploying a SAML based system. A more extensive discussion of the legal issues
171 related to privacy and best practices related to privacy may be found in the Liberty Privacy and Security
172 Best Practices document [LibBestPractices].

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

2.1 Ensuring Confidentiality

176

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

181 It is technically possible to convey information confidentially (a discussion of common methods for
182 providing confidentiality occurs in the Security portion of the document in Section 4.2). All parties to SAML-
183 enabled transactions should analyze each of their steps in the interaction (and any subsequent uses of
184 data obtained from the transactions) to ensure that information that should be kept confidential is actually
185 being kept so.

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

2.2 Notes on Anonymity

192

193 The following sections discuss the concept of anonymity.

2.2.1 Definitions That Relate to Anonymity

194

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

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

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

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

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

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

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

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

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

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

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

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

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

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

233 **2.2.2 Pseudonymity and Anonymity**

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

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

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

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

244 Only one-time-use pseudonyms can give you anonymity (within SAML, consider this as "anonymity within
245 a set").

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

250 **2.2.3 Behavior and Anonymity**

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

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

256 It was Klein's idiosyncratic usages that did him in (though apparently all authors have them).
257 The relevant point for SAML is that an "anonymous" user (even one that is never named) can be identified
258 enough to be harmed by repeated unusual behavior. Here are some examples:
259 • A user who each Tuesday at 21:00 access a database that correlates finger lengths and life span
260 starts to be non-anonymous. Depending on that user's other behavior, she or he may become
261 "traceable" [Pooling] in that other "identifying" information may be able to be collected.
262 • A user who routinely buys a usual set of products from a networked vending machine certainly
263 opens themselves to harm (by virtue of booby-trapping the products).

264 **2.2.4 Implications for Privacy**

265 Origin site authorities (such as authentication authorities and attribute authorities) can provide a degree of
266 "partial anonymity" by employing one-time-use identifiers or keys (for the "holder of key" case).
267 This anonymity is "partial" at best because the Subject is necessarily confined to the set of Subjects in a
268 relationship with the Authority.
269 This set may be further reduced (thus further reducing anonymity) when aggregating attributes are used
270 that further subset the user community at the origin site.
271 Users who truly care about anonymity must take care to disguise or avoid unusual patterns of behavior
272 that could serve to "de-anonymize" them over time.

273 3 Security

274 The following sections discuss security considerations.

275 3.1 Background

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

282 SAML is intended to aid deployers in establishing security contexts for application-level computer-based
283 communications within or between security domains. In this role, SAML transfers authentication data,
284 supporting end systems' ability to protect against unauthorized usage. Communications security is directly
285 applicable to the design of SAML. Systems security is of interest mostly in the context of SAML's threat
286 models. Section 2 of the IETF guidelines gives an overview of communications security and systems
287 security.

288 3.2 Scope

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

- 293 • Initial authentication: SAML allows statements to be made about acts of authentication that have
294 occurred, but includes no requirements or specifications for these acts of authentication.
295 Consumers of authentication assertions should be wary of blindly trusting these assertions
296 unless and until they know the basis on which they were made. Confidence in the assertions
297 must never exceed the confidence that the asserting party has correctly arrived at the
298 conclusions asserted.
- 299 • Trust Model: In many cases, the security of a SAML conversation will depend on the underlying
300 trust model, which is typically based on a key management infrastructure (for example, PKI or
301 secret key). For example, SOAP messages secured by means of XML Signature [XMLSig] are
302 secured only insofar as the keys used in the exchange can be trusted. Undetected compromised
303 keys or revoked certificates, for example, could allow a breach of security. Even failure to require
304 a certificate opens the door for impersonation attacks. PKI setup is not trivial and must be
305 implemented correctly in order for layers built on top of it (such as parts of SAML) to be secure.
- 306 • Suitable implementations of security protocols is necessary to maintain the security of a system,
307 including secure random or pseudo-random number generation and secure key storage.

308 3.3 SAML Threat Model

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

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

317 The following scenarios describe possible attacks:

- 318 • Collusion: The secret cooperation between two or more system entities to launch an attack, for
319 example:
- 320 Collusion between Principal and service provider
321 Collusion between Principal and identity provider
322 Collusion between identity provider and service provider
323 Collusion among two or more Principals
324 Collusion between two or more service providers
325 Collusion between two or more identity providers
- 326 • Denial-of-Service Attacks: The prevention of authorized access to a system resource or the
327 delaying of system operations and functions.
- 328 • Man-in-the-Middle Attacks: A form of active wiretapping attack in which the attacker intercepts
329 and selectively modifies communicated data to masquerade as one or more of the entities
330 involved in a communication association.
- 331 • Replay Attacks: An attack in which a valid data transmission is maliciously or fraudulently
332 repeated, either by the originator or by an adversary who intercepts the data and retransmits it,
333 possibly as part of a masquerade attack.
- 334 • Session Hijacking: A form of active wiretapping in which the attacker seizes control of a
335 previously established communication association.

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

340 The direct consequence of such a scoping is that the security of a system based on assertions as inputs is
341 only as good as the security of the system used to generate those assertions, and of the correctness of
342 the data and processing on which the generated assertions are based. When determining what issuers to
343 trust, particularly in cases where the assertions will be used as inputs to authentication or authorization
344 decisions, the risk of security compromises arising from the consumption of false but validly issued
345 assertions is a large one. Trust policies between asserting and relying parties should always be written to
346 include significant consideration of liability and implementations should provide an appropriate audit trail.

347 4 Security Techniques

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

350 4.1 Authentication

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

353 4.1.1 Active Session

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

359 4.1.2 Message-Level

360 XML Signature [XMLSig] and the OASIS Web Services Security specifications [WSS] provide methods of
361 creating a persistent “authentication” that is tightly coupled to a document. This method does not
362 independently guarantee that the sender of the message is in fact that signer (and indeed, in many cases
363 where intermediaries are involved, this is explicitly not the case).
364 Any method that allows the persistent confirmation of the involvement of a uniquely resolvable entity with a
365 given subset of an XML message is sufficient to meet this requirement.

366 4.2 Confidentiality

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

369 4.2.1 In Transit

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

372 4.2.2 Message-Level

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

375 4.3 Data Integrity

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

378 4.3.1 In Transit

379 Use of a secure network protocol such as TLS [RFC2246] or the IP Security Protocol [IPsec] may be
380 configured to provide integrity protection for the packets transmitted via the network connection.

381 4.3.2 Message-Level

382 XML Signature [XMLSig] provides a method of creating a persistent guarantee of the unaltered nature of a

383 message that is tightly coupled to that message.
384 Any method that allows the persistent confirmation of the unaltered nature of a given subset of an XML
385 message is sufficient to meet this requirement.

386 **4.4 Notes on Key Management**

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

391 **4.4.1 Access to the Key**

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

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

400 **4.4.2 Binding of Identity to Key**

401 For a key-based system to be used for authentication there must be some trusted binding of identity to
402 key. Verifying a digital signature on a document can determine if the document is unaltered since it was
403 signed, and that it was actually signed by a given key. However, this does not confirm that the key used is
404 actually the key of a specific individual appropriate for the time and purpose. Verifying the binding of a key
405 to a party requires additional validation.

406 This key-to-individual binding must be established. Common solutions include local directories that store
407 both identifiers and key—which is simple to understand but difficult to maintain—or the use of certificates.
408 Using certificates can provide a scalable means to associate a key with an identity, but requires
409 mechanisms to manage the certificate lifecycle and changes to the status of the binding (e.g. An
410 employee leaves and no longer has a corporate identity). One common approach is to use a Public Key
411 Infrastructure (PKI).

412 In this case a set of trusted root Certifying Authorities (CAs) are identified for each consumer of signatures
413 —answering the question “Whom do I trust to make statements of identity-to-key binding?” Verification of
414 a signature then becomes a process of first verifying the signature (to determine that the signature was
415 done by the key in question and that the message has not changed) and then validating the certificate
416 chain (to determine that the key is bound to the right identity) and validating that the binding is still
417 appropriate. Validating the binding requires steps to be taken to ensure that the binding is currently valid
418 —a certificate typically has a “lifetime” built into it, but if a key is compromised during the life of the
419 certificate then the key-to-identity binding contained in the certificate becomes invalid while the certificate
420 is still valid on its face. Also, certificates often depend on associations that may end before their lifetime
421 expires (for example, certificates that should become invalid when someone changes employers, etc.)
422 Different mechanisms may be used to validate key and certificate validity, such as Certificate Revocation
423 Lists (CRLs), the Online Certificate Status Protocol [OCSP], or the XML Key Management Specification
424 (XKMS) [XKMS], but these mechanisms are out of scope of the SSTC work.

425 A proper key management system is thus quite strong but very complex. Verifying a signature ends up
426 being a process of verifying the document-to-key binding, then verifying the key-to-identity binding, as well
427 as the current validity of the key and certificate.

428 **4.5 SSL/TLS Cipher Suites**

429 The use of HTTP over SSL 3.0 or TLS 1.0 [RFC2246] , or use of URLs with the HTTPS URL scheme, is
430 strongly recommended at many places in this document.

431 Unless otherwise specified, in any SAML binding's use of SSL 3.0 [SSL3] or TLS 1.0 [RFC2246], servers
432 MUST authenticate to clients using a X.509 v3 certificate. The client MUST establish server identity based
433 on contents of the certificate (typically through examination of the certificate's subject DN field).

434 SSL/TLS can be configured to use many different cipher suites, not all of which are adequate to provide
435 "best practices" security. The following sections provide a brief description of cipher suites and
436 recommendations for cipher suite selection.

437 4.5.1 SSL/TLS Cipher Suites

438 **Note:** While references to the US Export restrictions are now obsolete, the constants
439 naming the cipher suites have not changed. Thus,
440 SSL_DHE_DSS_EPORT_WITH_DES40_CBC_SHA is still a valid cipher suite identifier,
441 and the explanation of the historical reasons for the inclusion of "EXPORT" has been left
442 in place in the following summary.

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

- 448 • The protocol, SSL or TLS.
- 449 • The type of key exchange algorithm used. SSL defines many; the ones that provide server
450 authentication are the most important ones, but anonymous key exchange is supported. (Note
451 that anonymous key exchange algorithms are subject to "man in the middle" attacks, and are **not**
452 **recommended** in the SAML context.) The "RSA" authenticated key exchange algorithm is
453 currently the most interoperable algorithm. Another important key exchange algorithm is the
454 authenticated Diffie-Hellman "DHE_DSS" key exchange, which has no patent-related
455 implementation constraints.¹
- 456 • Whether the key exchange algorithm is freely exportable from the United States of America.
457 Exportable algorithms must use short (512-bit) public keys for key exchange and short (40-bit)
458 symmetric keys for encryption. Keys of these lengths have been successfully attacked, and their
459 use is not recommended.
- 460 • The encryption algorithm used. The fastest option is the RC4 stream cipher; DES and variants
461 (DES40, 3DES-EDE) as well as AES are also supported in "cipher block chaining" (CBC) mode.
462 Other modes are also supported, refer to the TLS documentation [RFC2246].
- 463 • Null encryption is also an option in some cipher suites. Note that null encryption performs **no**
464 encryption; in such cases SSL/TLS is used only to authenticate and provide integrity protection.
465 Cipher suites with null encryption do not provide confidentiality, and **must not be used** in cases
466 where confidentiality is a requirement and is not obtained by means other than SSL/TLS.
- 467 • The digest algorithm used for the Message Authentication Code. The recommended choice is
468 SHA1.
- 469 • For example, the cipher suite named SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
470 uses SSL, uses an authenticated Diffie-Hellman key exchange (DHE_DSS), is export grade
471 (EXPORT), uses an exportable variant of the DES cipher (DES40_CBC), and uses the SHA1
472 digest algorithm in its MAC (SHA).

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

1 ¹ The RSA algorithm patent has expired; hence this issue is mostly historical.

477 **4.5.2 SSL/TLS Recommendations**

478 SSL 2.0 must not be used due to known security weaknesses. TLS is preferred, SSL 3.0 may also be
479 used.

480 The SAML 2.0 Bindings specification outlines which cipher suites are required and recommended, making
481 normative statements. This section repeats this information for completeness, but that specification is
482 considered normative in case of inconsistency.

483 TLS-capable implementations MUST implement the TLS_RSA_WITH_3DES_EDE_CBC_SHA cipher
484 suite and MAY implement the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite.

485 FIPS [FIPS] TLS-capable implementations MUST implement the corresponding
486 TLS_RSA_FIPS_WITH_3DES_EDE_CBC_SHA cipher suite and MAY implement the corresponding
487 TLS_RSA_FIPS_AES_128_CBC_SHA cipher suite [FIPS].

488 SSL-capable implementations MUST implement the SSL_RSA_WITH_3DES_EDE_CBC_SHA cipher
489 suite.

490 FIPS [FIPS] SSL-capable implementations MUST implement the FIPS ciphersuite corresponding to the
491 SSL SSL_RSA_WITH_3DES_EDE_CBC_SHA cipher suite [FIPS].

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

- 495 • TLS_RSA_WITH_AES_128_CBC_SHA

5 General SAML Security Considerations

496

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

5.1 SAML Assertions

499

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

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

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

5.2 SAML Protocol

517

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

5.2.1 Denial of Service

520

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

5.2.1.1 Requiring Client Authentication at a Lower Level

527

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

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

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

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

540 **5.2.1.2 Requiring Signed Requests**

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

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

552 **5.2.1.3 Restricting Access to the Interaction URL**

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

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

6 SAML Bindings Security Considerations

559

560 The security considerations in the design of the SAML request-response protocol depend to a large extent
561 on the particular protocol binding (as defined in the SAML bindings specification [SAMLBind]) that is used.
562 The bindings sanctioned by the OASIS Security Services Technical Committee are the SOAP binding,
563 Reverse SOAP Binding (PAOS), HTTP Redirect binding, HTTP Redirect/POST binding and HTTP Artifact
564 binding and SAML URI bindings.

6.1 SAML SOAP Binding

565

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

6.1.1 Eavesdropping

570

571 **Threat:** Since there is no in-transit confidentiality requirement, it is possible that an eavesdropping party
572 could acquire both the SOAP message containing a request and the SOAP message containing the
573 corresponding response. This acquisition exposes both the nature of the request and the details of the
574 response, possibly including one or more assertions.

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

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

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

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

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

599 The details of how to provide this confidentiality depend on the specific SOAP transport chosen. Using
600 HTTP over TLS/SSL (described further in Section 6.1.7) is one method. Other transports will necessitate
601 other in-transit confidentiality techniques; for example, an SMTP transport might use S/MIME.

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

605 6.1.2 Replay

606 **Threat:** There is little vulnerability to replay attacks at the level of the SOAP binding. Replay is more of an
607 issue in the various profiles. The primary concern about replay at the SOAP binding level is the potential
608 for use of replay as a denial-of-service attack method.

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

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

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

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

625 6.1.3 Message Insertion

626 **Threat:** A fabricated request or response is inserted into the message stream. A false response such as
627 a spurious “yes” reply to an authorization decision query or the return of false attribute information in
628 response to an attribute query may result in inappropriate receiver action.

629 **Countermeasures:** The ability to insert a request is not a threat at the SOAP binding level. The threat of
630 inserting a false response can be a denial of service attack, for example returning SOAP Faults for
631 responses, but this attack would become quickly obvious. The more subtle attack of returning fabricated
632 responses is addressed in the SAML protocol, appropriate since according to the SOAP Binding definition
633 each SOAP response must contain a single SAML protocol response unless it contains a fault. The SAML
634 Protocol addresses this with two mechanisms, correlation of responses to requests using the required
635 InResponseTo attribute, making an attack harder since requests must be intercepted to generate
636 responses, and through the support origin authentication, either via signed SAML responses or through a
637 secured transport connection such as SSL/TLS.

638 6.1.4 Message Deletion

639 **Threat:** The message deletion attack would either prevent a request from reaching a responder, or would
640 prevent the response from reaching the requester.

641 **Countermeasures:** In either case, the SOAP binding does not address this threat. In general, correlation
642 of request and response messages may deter such an attack, for example use of the InResponseTo
643 attribute in the SAMLResponseType.

644 6.1.5 Message Modification

645 **Threat:** Message modification is a threat to the SOAP binding in both directions.

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

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

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

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

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

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

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

671 **6.1.6 Man-in-the-Middle**

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

675 **Countermeasures:** A bilateral authentication system would allow both parties to determine that what they
676 are seeing in a conversation actually came from the other party to the conversation.

677 At the SOAP binding level, this goal could also be accomplished by digitally signing both requests and
678 responses (with all the caveats discussed in Section 6.1.5 above). This method does not prevent an
679 eavesdropper from sitting in the middle and forwarding both ways, but he is prevented from altering the
680 conversation in any way without being detected.

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

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

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

691 **6.1.7 Use of SOAP over HTTP**

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

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

701 The HTTP Authentication RFC [RFC2617] describes possible attacks in the HTTP environment when
702 basic or message-digest authentication schemes are used.

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

707 **6.2 Reverse SOAP (PAOS) Binding**

708 **6.2.1 Denial of Service**

709 **Threat:** Remove HTTP accept header field and/or the PAOS HTTP header field causing HTTP responder
710 to ignore PAOS processing possibility.

711 **Countermeasures:** Integrity protect the HTTP message, using SSL/TLS integrity protection or other
712 adequate transport layer security mechanism.

713 **6.3 HTTP Redirect binding**

714 **6.3.1 Denial of Service**

715 **Threat:** Malicious redirects into identity or service provider targets

716 Description: A spurious entity could issue a redirect to a user agent so that the user agent would access a
717 resource that disrupts single sign-on. For example, an attacker could redirect the user agent to a logout
718 resource of a service provider causing the Principal to be logged out of all existing authentication
719 sessions.

720 **Countermeasures:** Access to resources that produce side effects could be specified with a transient
721 qualifier that must correspond to the current authentication session. Alternatively, a confirmation dialog
722 could be interposed that relies on a transient qualifier with similar semantics.

723 **6.4 HTTP Redirect/POST binding**

724 This section utilizes materials from [ShibMarlena and [Rescorla-Sec] and is derived from material in the
725 SAML 1.1 Bindings and Profiles specification [SAML11Bind].

726 **6.4.1 Stolen Assertion**

727 **Threat:** If an eavesdropper can copy the real user’s SAML response and included assertions, then the
728 eavesdropper could construct an appropriate POST body and be able to impersonate the user at the
729 destination site.

730 **Countermeasures:** Confidentiality MUST be provided whenever a response is communicated between a
731 site and the user’s browser. This provides protection against an eavesdropper obtaining a real user’s
732 SAML response and assertions.

733 If an eavesdropper defeats the measures used to ensure confidentiality, additional countermeasures are
734 available:

- 735 • The Identity Provider and Service Provider sites SHOULD make some reasonable effort to
736 ensure that clock settings at both sites differ by at most a few minutes. Many forms of time
737 synchronization service are available, both over the Internet and from proprietary sources.
- 738 • When a non-SSO SAML profile uses the POST binding it must ensure that the receiver can
739 perform timely subject confirmation. To this end, a SAML authentication assertion for the
740 principal MUST be included in the POSTed form response.
- 741 • Values for `NotBefore` and `NotOnOrAfter` attributes of SSO assertions SHOULD have the
742 shortest possible validity period consistent with successful communication of the assertion from
743 Identity Provider to Service Provider site. This is typically on the order of a few minutes. This
744 ensures that a stolen assertion can only be used successfully within a small time window.
- 745 • The Service Provider site MUST check the validity period of all assertions obtained from the

746 Identity Provider site and reject expired assertions. A Service Provider site MAY choose to
747 implement a stricter test of validity for SSO assertions, such as requiring the assertion's
748 `IssueInstant` or `AuthenticationInstant` attribute value to be within a few minutes of the
749 time at which the assertion is received at the Service Provider site.

750 • If a received authentication statement includes a `<saml:SubjectLocality>` element with the
751 IP address of the user, the Service Provider site MAY check the browser IP address against the
752 IP address contained in the authentication statement.

753 6.4.2 Man In the Middle Attack

754 **Threat:** Since the Service Provider site obtains bearer SAML assertions from the user by means of an
755 HTML form, a malicious site could impersonate the user at some new Service Provider site. The new
756 Service Provider site would believe the malicious site to be the subject of the assertion.

757 **Countermeasures:** The Service Provider site MUST check the `Recipient` attribute of the SAML response
758 to ensure that its value matches the `https://<assertion consumer host name and path>`. As the
759 response is digitally signed, the `Recipient` value cannot be altered by the malicious site.

760 6.4.3 Forged Assertion

761 **Threat:** A malicious user, or the browser user, could forge or alter a SAML assertion.

762 **Countermeasures:** The browser/POST profile requires the SAML response carrying SAML assertions to
763 be signed, thus providing both message integrity and authentication. The Service Provider site MUST
764 verify the signature and authenticate the issuer.

765 6.4.4 Browser State Exposure

766 **Threat:** The browser/POST profile involves uploading of assertions from the web browser to a Service
767 Provider site. This information is available as part of the web browser state and is usually stored in
768 persistent storage on the user system in a completely unsecured fashion. The threat here is that the
769 assertion may be "reused" at some later point in time.

770 **Countermeasures:** Assertions communicated using this profile must always have short lifetimes and
771 should have a `<OneTimeUse>` SAML assertion `<Conditions>` element. Service Provider sites are
772 expected to ensure that the assertions are not re-used.

773 6.4.5 Replay

774 **Threat:** Replay attacks amount to resubmission of the form in order to access a protected resource
775 fraudulently.

776 **Countermeasures:** The profile mandates that the assertions transferred have the one-use property at the
777 Service Provider site, preventing replay attacks from succeeding.

778 6.4.6 Modification or Exposure of state information

779 **Threat:** Relay state tampering or fabrication

780 Some of the messages may carry a `<RelayState>` element, which is recommended to be integrity-
781 protected by the producer and optionally confidentiality- protected. If these practices are not followed, an
782 adversary could trigger unwanted side effects. In addition, by not confidentiality-protecting the value of this
783 element, a legitimate system entity could inadvertently expose information to the identity provider or a
784 passive attacker.

785 **Countermeasure:** Follow the recommended practice of confidentiality- and integrity- protecting the
786 `RelayState` data. Note: Because the value of this element is both produced and consumed by the same
787 system entity, symmetric cryptographic primitives could be utilized

788 6.5 HTTP Artifact Binding

789 This section utilizes materials from [ShibMarlena and [Rescorla-Sec] and is derived from material in the
790 SAML 1.1 Bindings and Profiles specification [SAML11Bind].

791 6.5.1 Stolen Artifact

792 **Threat:** If an eavesdropper can copy the real user's SAML artifact, then the eavesdropper could construct
793 a URL with the real user's SAML artifact and be able to impersonate the user at the destination site.

794 **Countermeasures:** Confidentiality **MUST** be provided whenever an artifact is communicated between a
795 site and the user's browser. This provides protection against an eavesdropper gaining access to a real
796 user's SAML artifact.

797 If an eavesdropper defeats the measures used to ensure confidentiality, additional countermeasures are
798 available:

- 799 • The source and destination sites **SHOULD** make some reasonable effort to ensure that clock
800 settings at both sites differ by at most a few minutes. Many forms of time synchronization service
801 are available, both over the Internet and from proprietary sources.
- 802 • The source site **SHOULD** track the time difference between when a SAML artifact is generated
803 and placed on a URL line and when a `<samlp:Request>` message carrying the artifact is
804 received from the destination. A maximum time limit of a few minutes is recommended. Should
805 an assertion be requested by a destination site query beyond this time limit, the source site
806 **MUST** not provide the assertions to the destination site.
- 807 • It is possible for the source site to create SSO assertions either when the corresponding SAML
808 artifact is created or when a `<samlp:Request>` message carrying the artifact is received from
809 the destination. The validity period of the assertion **SHOULD** be set appropriately in each case:
810 longer for the former, shorter for the latter.
- 811 • Values for `NotBefore` and `NotOnOrAfter` attributes of SSO assertions **SHOULD** have the
812 shortest possible validity period consistent with successful communication of the assertion from
813 source to destination site. This is typically on the order of a few minutes. This ensures that a
814 stolen artifact can only be used successfully within a small time window.
- 815 • The destination site **MUST** check the validity period of all assertions obtained from the source
816 site and reject expired assertions. A destination site **MAY** choose to implement a stricter test of
817 validity for SSO assertions, such as requiring the assertion's `IssueInstant` or
818 `AuthenticationInstant` attribute value to be within a few minutes of the time at which the
819 assertion is received at the destination site.
- 820 • If a received authentication statement includes a `<saml:SubjectLocality>` element with the
821 IP address of the user, the destination site **MAY** check the browser IP address against the IP
822 address contained in the authentication statement.

823 6.5.2 Attacks on the SAML Protocol Message Exchange

824 **Threat:** The message exchange used by the Service Provider to obtain an assertion from the Identity
825 Provider could be attacked in a variety of ways, including artifact or assertion theft, replay, message
826 insertion or modification, and MITM (man-in-the-middle attack).

827 **Countermeasures:** The requirement for the use of a SAML protocol binding with the properties of
828 bilateral authentication, message integrity, and confidentiality defends against these attacks.

829 6.5.3 Malicious Destination Site

830 **Threat:** Since the Service Provider obtains artifacts from the user, a malicious site could impersonate the
831 user at some new Service Provider site. The new Service Provider site would obtain assertions from the
832 Identity Provider site and believe the malicious site to be the user.

833 **Countermeasures:** The new Service Provider site will need to authenticate itself to the Identity Provider

834 site so as to obtain the SAML assertions corresponding to the SAML artifacts. There are two cases to
835 consider:

- 836 1. If the new Service Provider site has no relationship with the Identity Provider site, it will be unable to
837 authenticate and this step will fail.
- 838 2. If the new Service Provider site has an existing relationship with the Identity Provider site, the
839 Identity Provider site will determine that assertions are being requested by a site other than that to
840 which the artifacts were originally sent. In such a case, the Identity Provider site MUST not provide
841 the assertions to the new Service Provider site.

842 **6.5.4 Forged SAML Artifact**

843 **Threat:** A malicious user could forge a SAML artifact.

844 **Countermeasures:** The Bindings specification provides specific recommendations regarding the
845 construction of a SAML artifact such that it is infeasible to guess or construct the value of a current, valid,
846 and outstanding assertion handle. A malicious user could attempt to repeatedly “guess” a valid SAML
847 artifact value (one that corresponds to an existing assertion at a Identity Provider site), but given the size
848 of the value space, this action would likely require a very large number of failed attempts. An Identity
849 Provider site SHOULD implement measures to ensure that repeated attempts at querying against non-
850 existent artifacts result in an alarm.

851 **6.5.5 Browser State Exposure**

852 **Threat:** The SAML browser/artifact profile involves “downloading” of SAML artifacts to the web browser
853 from an Identity Provider site. This information is available as part of the web browser state and is usually
854 stored in persistent storage on the user system in a completely unsecured fashion. The threat here is that
855 the artifact may be “reused” at some later point in time.

856 **Countermeasures:** The “one-use” property of SAML artifacts ensures that they cannot be reused from a
857 browser. Due to the recommended short lifetimes of artifacts and mandatory SSO assertions, it is difficult
858 to steal an artifact and reuse it from some other browser at a later time.

859 **6.5.6 Replay**

860 **Threat:** Reuse of an artifact by repeating protocol messages

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

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

866 **6.6 SAML URI Binding**

867 **6.6.1 Substitution**

868 **Threat:** Substitution of assertion with another by substitution of URI reference. Given that a URI is
869 opaque to the receiver it is hard to validate the integrity.

870 **Countermeasures:** Where this is a concern, transport layer integrity protection such as with SSL/TLS is
871 required.

872 7 SAML Profile Security Considerations

873 The SAML profiles specification [SAMLProf] defines profiles of SAML, which are sets of rules describing
874 how to embed SAML assertions into and extract them from a framework or protocol.

875 7.1 Web Browser Single Sign-On (SSO) Profiles

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

881 7.1.1 SSO Profile

882 7.1.1.1 Eavesdropping

883 **Threat:** The possibility of eavesdropping exists in all web browser cases.

884 **Countermeasures:** In cases where confidentiality is required (bearing in mind that any assertion that is
885 not sent securely, along with the requests associated with it, is available to the malicious eavesdropper),
886 HTTP traffic needs to take place over a transport that ensures confidentiality. HTTP over TLS/SSL
887 [RFC2246] and the IP Security Protocol [IPsec] meet this requirement.

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

889 7.1.1.2 Theft of the User Authentication Information

890 **Threat:** In the case where the subject authenticates to the source site by revealing reusable
891 authentication information, for example, in the form of a password, theft of the authentication information
892 will enable an adversary to impersonate the subject.

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

898 7.1.1.3 Theft of the Bearer Token

899 **Threat:** In the case where the authentication assertion contains the assertion bearer's authentication
900 protocol identifier, theft of the artifact will enable an adversary to impersonate the subject.

901 **Countermeasures:** Each of the following methods decreases the likelihood of this happening:

- 902 • The destination site implements a confidentiality safeguard on its connection with the subject's
903 browser.
- 904 • The subject or destination site ensures (out of band) that the source site implements a
905 confidentiality safeguard on its connection with the subject's browser.
- 906 • The destination site verifies that the subject's browser was directly redirected by a source site
907 that directly authenticated the subject.
- 908 • The source site refuses to respond to more than one request for an assertion corresponding to
909 the same assertion ID.
- 910 • If the assertion contains a condition element of type **AudienceRestrictionType** that identifies a
911 specific domain, then the destination site verifies that it is a member of that domain.

- 912 • The connection between the destination site and the source site, over which the assertion ID is
913 passed, is implemented with a confidentiality safeguard.
- 914 • The destination site, in its communication with the source site, over which the assertion ID is
915 passed, must verify that the source site is genuinely the expected and trusted source site.

916 **7.1.1.4 Replay**

917 The possibility of a replay attack exists for this set of profiles. A replay attack can be used either to attempt
918 to deny service or to retrieve information fraudulently. The specific countermeasures depend on which
919 specific binding is used and are discussed above

920 **7.1.1.5 Message Insertion**

921 Message insertion attacks are discussed in the section on bindings.

922 **7.1.1.6 Message Deletion**

923 **Threat:** Deleting a message during any step of the interactions between the browser, SAML assertion
924 issuer, and SAML assertion consumer will cause the interaction to fail. It results in a denial of some
925 service but does not increase the exposure of any information.

926 **Countermeasures:** Use of an integrity protected transport channel addresses the threat of message
927 deletion when no intermediaries are present.

928 **7.1.1.7 Message Modification**

929 **Threat:** The possibility of alteration of the messages in the stream exists for this set of profiles. Some
930 potential undesirable results are as follows:

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

936 **Countermeasures:**

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

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

941 **7.1.1.8 Man-in-the-Middle**

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

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

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

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

956 **7.1.1.9 Impersonation without Reauthentication**

957 **Threat:** Rogue user attempts to impersonate currently logged-in legitimate Principal and thereby gain
958 access to protected resources.

959 Once a Principal is successfully logged into an identity provider, subsequent <AuthnRequest> messages
960 from different service providers concerning that Principal will not necessarily cause the Principal to be
961 reauthenticated. Principals must, however, be authenticated unless the identity provider can determine
962 that an <AuthnRequest> is associated not only with the Principal's identity, but also with a validly
963 authenticated identity provider session for that Principal.

964 **Countermeasures:** In implementations where this threat is a concern, identity providers MUST maintain
965 state information concerning active sessions, and MUST validate the correspondence between an
966 <AuthnRequest> and an active session before issuing an <AuthnResponse> without first
967 authenticating the Principal. Cookies posted by identity providers MAY be used to support this validation
968 process, though Liberty does not mandate a cookie-based approach.

969 **7.1.2 Enhanced Client and Proxy Profile**

970 **7.1.2.1 Man in the Middle**

971 **Threat:** Intercept AuthnRequest and AuthnResponse SOAP messages, allowing subsequent Principal
972 impersonation.

973 A spurious system entity can interject itself as a man-in-the-middle (MITM) between the enhanced client
974 and a legitimate service provider, where it acts in the service provider role in interactions with the
975 enhanced client and in the enhanced client role in interactions with the legitimate service provider. In this
976 way, as a first step, the MITM is able to intercept the service provider's AuthnRequest and substitute any
977 URL of its choosing for the responseConsumerServiceURL value in the PAOS header block before
978 forwarding the AuthnRequest on to the enhanced client. Typically, the MITM will insert a URL value that
979 points back to itself. Then, if the enhanced client subsequently receives an AuthnResponse from the
980 identity provider and subsequently sends the contained AuthnResponse to the
981 responseConsumerServiceURL received from the MITM, the MITM will be able to masquerade as the
982 Principal at the legitimate service provider.

983 **Countermeasure:** The identity provider specifies to the enhanced client the address to which the
984 enhanced client must send the :AuthnResponse. The responseConsumerServiceURL in the PAOS
985 header is only used for error responses from the enhanced client – as specified in the profile.

986 **7.1.2.2 Denial of Service**

987 **Threat:** Change an AuthnRequest SOAP request so that it cannot be processed, such as by changing
988 the PAOS header block service attribute value to an unknown value or by changing the ECP header block
989 ProviderID or IDPList to cause the request to fail.

990 **Countermeasures:** Provide integrity protection for the SOAP message, by using SOAP Message Security
991 or SSL/TLS.

992 **7.1.3 Identity Provider Discovery Profile**

993 **Threat:** Cookie poisoning attack, where parameters within the cookie are modified, to cause discovery of
994 an fraudulent identity provider for example.

995 **Countermeasures:** The specific mechanism of using a common domain limits the feasibility of this threat.

996 **7.1.4 Single Logout Profile**

997 **Threat:** Passive attacker can collect a Principal's name identifier

998 During the initial steps, a passive attacker can collect the <LogoutRequest> information when it is issued
999 in the redirect. Exposing these data poses a privacy threat.

1000 **Countermeasures:** All exchanges should be conducted over a secure transport such as SSL or TLS.

1001 **Threat:** Unsigned <LogoutRequest> message

1002 An Unsigned <LogoutRequest> could be injected by a spurious system entity thus denying service to
1003 the Principal. Assuming that the NameIdentifier can be deduced or derived then it is conceivable that the
1004 user agent could be directed to deliver a fabricated <LogoutRequest> message.

1005 **Countermeasures:** Sign the <LogoutRequest> message. The identity provider can also verify the
1006 identity of a Principal in the absence of a signed request.

1007 **7.2 Name Identifier Management Profiles**

1008 **Threat:** Allow system entities to correlate information or otherwise inappropriately expose identity
1009 information, harming privacy.

1010 **Countermeasures:** IDP must take care to use different name identifiers with different service providers
1011 for same principal. The IDP SHOULD encrypt the name identifier it returns to the service provider,
1012 allowing subsequent interactions to use an opaque identifier.

1013 **7.3 Attribute Profiles**

1014 Threats related to bindings associated with attribute profiles are discussed above. No additional profile-
1015 specific threats are known.

1016

8 Summary

1017 Security and privacy must be addressed in a systemic manner, considering human issues such as social
1018 engineering attacks, policy issues, key management and trust management, secure implementation and
1019 other factors outside the scope of this document. Security technical solutions have a cost, so
1020 requirements and policy alternatives must also be considered, as must legal and regulatory requirements.

1021 This non-normative document summarizes general security issues and approaches as well as specific
1022 threats and countermeasures for the use of SAML assertions, protocols, bindings and profiles in a secure
1023 manner that maintains privacy. Normative requirements are specified in the normative SAML
1024 specifications.

9 References

1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074

The following are cited in the text of this document:

- [Anonymity]** Anonymity, Unobservability, and Pseudonymity – A Proposal for Terminology
Andreas Pfitzmann, Marit Köhntopp,
http://www.realname-diskussion.info/anon_terminology.pdf.
- [FIPS]** FIPS SSL CipherSuites, <http://www.mozilla.org/projects/security/pki/nss/ssl/fips-ssl-ciphersuites.html>.
- [FreeHaven]** The Free Haven Project: Distributed Anonymous Storage Service
Roger Dingledine & Michael J. Freedman & David Molnar
<http://www.freehaven.net/paper/node6.html>
<http://www.freehaven.net/paper/node7.html>
- [IPsec]** IETF IP Security Protocol Working Group, <http://www.ietf.org/html.charters/ipsec-charter.html>.
- [LibBestPractices]** C. Varney et al, Privacy and Security Best Practices, Version 2.0, November 12, 2003,
http://www.projectliberty.org/specs/final_privacy_security_best_practices.pdf
- [OCSP]** "X.509 Internet Public Key Infrastructure – Online Certificate Status Protocol - OCSP," M. Myers, et al., IETF RFC 2560, June 1999, <http://ietf.org/rfc/rfc2560.txt>
- [Pooling]** Pooling Intellectual Capital: Thoughts on Anonymity, Pseudonymity, and Limited Liability in Cyberspace
David G. Post
<http://www.cli.org/DPost/paper8.htm>
- [Rescorla-Sec]** E. Rescorla et al., *Guidelines for Writing RFC Text on Security Considerations*, Best Current Practice RFC 3552, July 2003,
<http://www.ietf.org/rfc/rfc3552.txt?number=3552>
- [RFC2246]** The TLS Protocol Version 1.0, <http://www.ietf.org/rfc/rfc2246.html>.
- [RFC2617]** J. Franks et al, HTTP Authentication: Basic and Digest Access Authentication, RFC 2617, <http://www.ietf.org/rfc/rfc2617.txt>
- [SAML11Bind]** Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML) V1.1, OASIS Standard, 2 September 2003 <http://www.oasis-open.org/committees/download.php/3405/oasis-%20sstc-saml-bindings-1.1.pdf>
- [SAMLBind]** S. Cantor et al., *Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS SSTC, January 2005. Document ID sstc-saml-bindings-2.0-cd-04. See <http://www.oasis-open.org/committees/security/>.
- [SAMLCore]** S. Cantor et al., *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS SSTC, January 2005. Document ID sstc-saml-core-2.0-cd-04. See <http://www.oasis-open.org/committees/security/>.
- [SAMLGloss]** J. Hodges et al., *Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS SSTC, January 2005. Document ID sstc-saml-glossary-2.0-cd-04. See <http://www.oasis-open.org/committees/security/>.
- [SAMLProf]** S. Cantor et al., *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS SSTC, January 2005. Document ID sstc-saml-profiles-2.0-cd-04. See <http://www.oasis-open.org/committees/security/>.
- [ShibMarlena]** Marlena Erdos, Shibboleth Architecture DRAFT v1.1,
<http://shibboleth.internet2.edu/draft-internet2-shibboleth-arch-v05.html>.
- [SRMPres]** Message Queuing: Messaging Over The Internet
Shai Kariv
<http://www.microsoft.com/israel/events/teched/presentations/EN308.zip>
- [SSL3]** "The SSL Protocol Version 3.0", <http://wp.netscape.com/eng/ssl3/draft302.txt>
- [WSS]** Web Services Security specifications (WSS), OASIS. <http://www.oasis->

- 1075 [open.org/committees/wss](http://www.oasis-open.org/committees/wss).
- 1076 **[WSS-SAML]** P. Hallam-Baker et al., *Web Services Security: SAML Token Profile*, OASIS,
1077 March 2003, <http://www.oasis-open.org/committees/wss>.
- 1078 **[XKMS]** “XML Key Management Specifications (XKMS 2.0)”, W3C Candidate
1079 Recommendation, 5 April 2004, <http://www.w3.org/TR/xkms2/>
- 1080 **[XMLEnc]** Donald Eastlake et al., *XML Encryption Syntax and Processing*,
1081 <http://www.w3.org/TR/xmlenc-core/>, World Wide Web Consortium, December
1082 2002.
- 1083 **[XMLSig]** Donald Eastlake et al., *XML-Signature Syntax and Processing*,
1084 <http://www.w3.org/TR/xmlsig-core/>, World Wide Web Consortium.
- 1085 The following additional documents are recommended reading:
- 1086 **[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)
1087 [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
1088 the security module is the material of interest.
- 1089 **[ebXML-Risk]** ebXML Technical Architecture Risk Assessment v1.0,
1090 <http://www.ebxml.org/specs/secRISK.pdf>.
- 1091 **[Prudent]** Prudent Engineering Practice for Cryptographic Protocols,
1092 <http://citeseer.nj.nec.com/abadi96prudent.html>.
- 1093 **[Robustness]** Robustness principles for public key protocols,
1094 <http://citeseer.nj.nec.com/2927.html>.

1095 Appendix A. Acknowledgments

1096 The editors would like to acknowledge the contributions of the OASIS Security Services Technical
1097 Committee, whose voting members at the time of publication were:

- 1098 • Conor Cahill, AOL
- 1099 • John Hughes, Atos Origin
- 1100 • Hal Lockhart, BEA Systems
- 1101 • Mike Beach, Boeing
- 1102 • Rebekah Metz, Booz Allen Hamilton
- 1103 • Rick Randall, Booz Allen Hamilton
- 1104 • Ronald Jacobson, Computer Associates
- 1105 • Carolina Canales-Valenzuela, Ericsson
- 1106 • Dana Kaufman, Forum Systems
- 1107 • Irving Reid, Hewlett-Packard
- 1108 • Paula Austel, IBM
- 1109 • Michael McIntosh, IBM
- 1110 • Anthony Nadalin, IBM
- 1111 • Nick Ragouzis, Individual
- 1112 • Scott Cantor, Internet2
- 1113 • Bob Morgan, Internet2
- 1114 • Peter Davis, Neustar
- 1115 • Jeff Hodges, Neustar
- 1116 • Frederick Hirsch, Nokia
- 1117 • Senthil Sengodan, Nokia
- 1118 • Abbie Barbir, Nortel Networks
- 1119 • Scott Kiestler, Novell
- 1120 • Cameron Morris, Novell
- 1121 • Paul Madsen, NTT
- 1122 • Steve Anderson, OpenNetwork
- 1123 • Ari Kermaier, Oracle
- 1124 • Vamsi Motukuru, Oracle
- 1125 • Darren Platt, Ping Identity
- 1126 • Prateek Mishra, Principal Identity
- 1127 • Jim Lien, RSA Security
- 1128 • John Linn, RSA Security
- 1129 • Rob Philpott, RSA Security
- 1130 • Dipak Chopra, SAP
- 1131 • Jahan Moreh, Sigaba
- 1132 • Bhavna Bhatnagar, Sun Microsystems
- 1133 • Eve Maler, Sun Microsystems
- 1134 • Ronald Monzillo, Sun Microsystems
- 1135 • Emily Xu, Sun Microsystems
- 1136 • Greg Whitehead, Trustgenix

1137 The editors also would like to acknowledge the following people for their contributions to previous versions
1138 of the OASIS Security Assertions Markup Language Standard:

- 1139 • Stephen Farrell, Baltimore Technologies
- 1140 • David Orchard, BEA Systems
- 1141 • Krishna Sankar, Cisco Systems
- 1142 • Zahid Ahmed, CommerceOne
- 1143 • Carlisle Adams, Entrust
- 1144 • Tim Moses, Entrust
- 1145 • Nigel Edwards, Hewlett-Packard
- 1146 • Joe Pato, Hewlett-Packard
- 1147 • Bob Blakley, IBM
- 1148 • Marlena Erdos, IBM
- 1149 • Marc Chanliau, Netegrity
- 1150 • Chris McLaren, Netegrity
- 1151 • Lynne Rosenthal, NIST
- 1152 • Mark Skall, NIST
- 1153 • Simon Godik, Overxeer
- 1154 • Charles Norwood, SAIC
- 1155 • Evan Prodromou, Securant
- 1156 • Robert Griffin, RSA Security (former editor)
- 1157 • Sai Allarvarpu, Sun Microsystems
- 1158 • Chris Ferris, Sun Microsystems
- 1159 • Mike Myers, Traceroute Security
- 1160 • Phillip Hallam-Baker, VeriSign (former editor)
- 1161 • James Vanderbeek, Vodafone
- 1162 • Mark O'Neill, Vordel
- 1163 • Tony Palmer, Vordel

1164 Finally, the editors wish to acknowledge the following people for their contributions of material used as
1165 input to the OASIS Security Assertions Markup Language specifications:

- 1166 • Thomas Gross, IBM
- 1167 • Birgit Pfitzmann, IBM

1168

Appendix B. Notices

1169 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
1170 might be claimed to pertain to the implementation or use of the technology described in this document or
1171 the extent to which any license under such rights might or might not be available; neither does it represent
1172 that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to
1173 rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made
1174 available for publication and any assurances of licenses to be made available, or the result of an attempt
1175 made to obtain a general license or permission for the use of such proprietary rights by implementors or
1176 users of this specification, can be obtained from the OASIS Executive Director.

1177 OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or
1178 other proprietary rights which may cover technology that may be required to implement this specification.
1179 Please address the information to the OASIS Executive Director.

1180 **Copyright © OASIS Open 2005. All Rights Reserved.**

1181 This document and translations of it may be copied and furnished to others, and derivative works that
1182 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
1183 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and
1184 this paragraph are included on all such copies and derivative works. However, this document itself may
1185 not be modified in any way, such as by removing the copyright notice or references to OASIS, except as
1186 needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights
1187 defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it
1188 into languages other than English.

1189 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
1190 or assigns.

1191 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1192 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
1193 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR
1194 ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.