Security Challenges, Threats and Countermeasures Version 1.0

Final Material

Date: 2005/05/07

This version:

http://www.ws-i.org/Profiles/BasicSecurity/SecurityChallenges-1.0-20050507.doc

Latest version:

http://www.ws-i.org/Profiles/BasicSecurity/SecurityChallenges-1.0.doc

Editors:
Jerry Schwarz, Oracle
Bret Hartman, DataPower
Anthony Nadalin, IBM
Chris Kaler, Microsoft
Mark Davis, Sarvega
Frederick Hirsch, Nokia Corporation
K. Scott Morrison, Layer 7

Copyright

Administrative contact:

secretary@ws-i.org

Table of Contents

1 Introduction 4
2 Glossary 5
2.1 Basic Definitions 5
2.1.1 Discussion 5
2.2 Messages 5
2.2.1 Discussion 6
2.3 SOAP 1.2 6
2.3.1 Discussion 7
2.4 Sending Messages 7
2.4.1 Discussion 7
3 Security Challenges 8

3.1 C-01: Peer Identification and Authentication 8
3.2 C-02: Data Origin Identification and Authentication 9
C-03: Data Integrity 10
3.2.1 C-03A: Transport Data Integrity 10
3.2.2 C-03B: SOAP Message Integrity 10
3.3 C-04: Data Confidentiality 11
3.3.1 C-04A: Transport Data Confidentiality 11
3.3.2 C–04B: SOAP message confidentiality 12
3.4 C-05: Message Uniqueness 12
4 Threats 14
5 Security Solutions, Mechanisms and Countermeasures 16
5.1 Transport Layer Security Descriptions 16
5.1.1 Integrity 17
5.1.2 Confidentiality 17
5.1.3 Authentication by HTTP Service 18
5.1.4 Authentication by HTTP User Agent 18
5.1.5 Attributes 19
5.1.6 Combinations 19
5.2 SOAP Message Layer Security Descriptions 20
5.2.1 Integrity 21
5.2.2 Confidentiality 21
5.2.3 SOAP Sender Authentication 21
5.2.4 Attributes 22
5.2.5 Message Uniqueness 22
5.2.6 Combinations 24
5.3 Combining Transport Layer and SOAP Message Layer Mechanisms 25
5.4 Transport and Message Layer Security Combinations 26
5.5 Security Considerations for Combinations 28
5.5.1 Transport Layer Security Solutions 28
5.5.2 SOAP Message Layer Security Solutions 31
5.5.3 Hybrid Security Solutions 32
6 Scenarios 34
6.1 Notation for Describing Scenarios 34
6.2 Conventions for Describing Security Requirements and Solutions 35
6.3 Terminology 35
6.4 Generic Security Requirements 35
6.4.1 Requirement: Peer Authentication 35
6.4.2 Requirement: Origin Authentication 36
6.4.3 Requirement: Integrity 36
6.4.4 Requirement: Confidentiality 36
6.4.5 Requirement: Message Uniqueness 37
6.5 Scenario Descriptions 37
6.5.1 Scenario: One-Way 37
6.5.2 Scenario: Synchronous Request/Response 38
6.5.3 Basic Callback 38
7 Out of Scope 41
7.1 Security Challenges 41
7.1.1 C-05: Non-Repudiation 41
7.1.2 C-06: Credentials Issuance 41
7.2 Threats 41
8 Acronyms 46

9 References 47
10 Informative References 48

1 Introduction

This document defines the requirements for and scope of the WS-I Basic Security Profile. The document is aimed at Web Services architects and developers who are examining the security aspects of the Web Services they are designing/developing.

This document:

- Identifies security challenges. These are general security goals or features that inform the selection of specific security requirements in scenarios.
- Identifies the typical threats that prevent accomplishment of each challenge.
- Identifies the typical countermeasures (technologies and protocols) used to mitigate each threat.
- Documents potential usage scenarios and the security challenges and threats that might apply to each (derived from the templates found in the Supply Chain Management Use Cases and WS-I Usage Scenarios documents).

This document assumes that the reader has at least a basic background in security technologies such as SSL/TLS, XML encryption and digital signatures, and OASIS Web Services Security [WSS 1.0]. It also assumes that the reader has a basic background in the message level technologies of SOAP.

.

2 Glossary

2.1 Basic Definitions

This section defines vocabulary that will be used to refer to the various entities and concepts in this document.

The following terms are used to describe certain entities.

- **Participant**: Any entity that plays some part in the scenarios. This is deliberately vague. No attempt is made to define entities or to characterize them. A participant might be a person, an institution, a computer, and a network or belong to some other category. Most obviously it includes the systems that exchange SOAP messages, but it also includes entities such as the original creator of content, or HTTP proxies that are not explicitly named in the scenarios.
- **SOAP Node**: [Copied with modification from [SOAP 1.1] The embodiment of the processing logic necessary to transmit, receive, process and/or relay a SOAP message, according to the set of conventions defined by SOAP 1.1 or SOAP 1.2. A SOAP node is responsible for enforcing the rules that govern the exchange of SOAP messages. It accesses the services provided by the underlying protocols through one or more SOAP bindings.

2.1.1 Discussion

An alternative is to use “entity” as the most abstract term and reserve “participant” for the SOAP nodes that are parts of scenarios. However, “entity” sounds a bit stilted. Note that a SOAP node is a participant.

2.2 Messages

Communication channels are inevitably layered. When, as in this document, it is necessary to discuss the interaction between layers some care is required to distinguish between events and messages at one level from those that occur at a lower level. In general what appears to be an atomic action, such as message transmission, at one level will have a more complicated structure at a lower level.

We are primarily interested in transmission of SOAP messages and the participants in the transmission. However in some cases we are also interested in non-SOAP messages.

- **Message**: Protocol elements that are exchanged, usually over a network, to affect a Web service (i.e. SOAP/HTTP messages)
- **SOAP Message**: [Copied from [SOAP 1.2] The basic unit of communication between SOAP nodes.

  Clarification: when using “SOAP with Attachments” [SwA] the attachments are considered part of the SOAP Message.
- **SOAP Layer**: The communication layer at which SOAP nodes reside.
- **HTTP Message**: The basic unit of HTTP communication, as defined in RFC 2616.
- **Transport Layer:** The communication layers below the SOAP layer.

- **SSL/TLS**: The communication layer below HTTP where security concerns are addressed See [RFC 2246]. There are technical differences between TLS and SSL, but these differences are not significant for this document. SSL/TLS refers to the profiled choice of SSL/TLS technology produced by the Basic Security Profile work group, and may thus be limited to versions of the technology as well as selected cipher suites and other profiling recommendations.
- **HTTPS**: The combination of HTTP with SSL/TLS.

2.2.1 Discussion

Normally HTTP and SSL/TLS would be considered separate layers. Consolidating them and lower layers compresses the stack. But it is convenient to treat HTTP, SSL/TLS and lower layers together.

2.3 SOAP 1.2

SOAP 1.2 defines the following terms:

- SOAP
- SOAP node
- SOAP role
- SOAP binding
- SOAP feature
- SOAP module
- SOAP message exchange pattern
- SOAP application
- SOAP message
- SOAP envelope
- SOAP header
- SOAP header block
- SOAP body
- SOAP fault
- SOAP sender
- SOAP receiver
- SOAP message path
- Initial SOAP sender
- SOAP intermediary
- Ultimate SOAP receiver.

2.3.1 Discussion

We adopt these terms with the understanding that we will apply them to SOAP 1.1 messages rather than SOAP 1.2 messages. We will not use any terms that refer specifically to SOAP 1.2 features that are not present in SOAP 1.1

2.4 Sending Messages

The participants in a message event are referred to as

- **Sender**: [From [BP 1.0]] The software that generates a message according to the protocol(s) associated with it.
- **Receiver**: [From [BP 1.0]] The software that consumes a message according to the protocol(s) associated with it (e.g. SOAP processors).

In most contexts it is not necessary to distinguish the various layers in the communication, however when it is necessary to do so “sender” or “receiver” may be modified by the protocol involved, so that “SOAP sender” and “HTTP receiver” can be used.

2.4.1 Discussion

The use of “sender” and “receiver” is so natural that it would be hard to avoid them even if they weren’t part of the official glossary.

3 Security Challenges

This section identifies potential security challenges that scenarios may want to address. The following subsections characterize the identified security challenges with the following attributes:

- ID: A unique challenge identifier in the form C-*nn*.
- Definition(s): One or more relevant definitions related to this challenge taken from the Internet Security Glossary [RFC 2828]
- Explanation: Supporting web services contextual explanation and comments. With further review and development, some explanations may be suitable as input to a WS-I Glossary that lists security-specific terms.
- Candidate technology: Technology solutions that can be used to address security threats and risks associated with this challenge. The suitability of a candidate technology is discussed in the discussion of each specific scenario, taking into account considerations for that scenario.
- Threat association: A mapping of security threats associated with the challenge, with references to specific threats outlined in Section 4 and Section 7.2. Threats that are related specifically to the provided explanation are included within the threat association. Threats that relate to the underlying mechanisms that are needed to address the security challenge are not identified. For example the exchange of authentication data should leverage integrity and confidentiality mechanisms; however, specific integrity and confidentiality threats are not identified for authentication challenges.
  Threats enumerated in Section 4 are labeled T-XX. Those in Section 7.2 are considered "out of scope" and labeled T(OOS)-XX. "Out of Scope" means they are not addressed by any available candidate technology. There is no connection between the numbering of these two groups.

3.1 C-01: Peer Identification and Authentication

Definitions:

Peer entity authentication: The corroboration that a peer entity in an association is the one claimed.

Identification: An act or process that presents an identifier to a system so that the system can recognize a system entity[1] and distinguish it from other entities.

Explanation: Any relationship between entities can be considered an "association" for purposes of this definition. For example, it does not require that the two entities directly communicate with each other.

Although the term "authentication" is sometimes used to include both the presentation and the corroboration of an identifier this document uses "authentication" in the narrower sense defined here.

A participant may convey information to another participant to establish identity in conjunction with the use of techniques to corroborate that information. The two SOAP participants are not necessarily directly connected by a single hop, for example the participants might be the initial

[1] Note that *System Entity*, used throughout this document, refers to the definition in RFC 2828.

SOAP sender and a second SOAP intermediary. Depending on application requirements (security policy) it may be reasonable to authenticate the sender, receiver or to use mutual authentication.

NOTE:

It is important for a relying party to ensure the correctness of the identification associated with authentication. For example, in using SSL/TLS a server may present an X.509 certificate to associate identity information with a public key and use the corresponding private key to prove possession of the private key. A relying party should not only rely on the authentication technology, but should also ensure that the information associated with the authentication is correct, thus authorizing further processing based on that information. This may include steps such as ensuring that the HTTP request domain name corresponds to the server certificate name and performing certificate validation. Such care is necessary in light of man-in-the-middle, DNS or TCP/IP attacks (T-04) where authentication may work technically but does not corroborate the correct party. Authorization is important but not addressed in this document.

Candidate technology:

- HTTPS with X.509 server authentication
- HTTP client authentication (Basic or Digest)
- HTTPS with X.509 mutual authentication of server and user agent
- OASIS SOAP Message Security

Threat association:

T-03, T-04, T-05, T-06, T-07, T(OOS)-01, T(OOS)-03, T(OOS)-04, T(OOS)-08, T(OOS)-13, T(OOS)-14.

3.2 C-02: Data Origin Identification and Authentication

Definitions:

Data origin authentication: The corroboration that the source of data received is as claimed.

Identification: An act or process that presents an identifier to a system so that the system can recognize a system entity and distinguish it from other entities.

Explanation: The provision and authentication of a declaration, carried in a web service message that some entity vouches for certain parts of the message. Note that it is possible that more than one entity might be involved in vouching for message parts. Also note that it is application-dependent as to how it is determined who initially created the message, as the message originator might be independent of, or hidden behind a vouching entity. This mechanism does not provide for the authentication of the destination prior to transmission of application data. However, the encryption of the data with a key only known to the legitimate destination can effectively serve as an implicit form of destination authentication if that is required.

This of course does not prevent the impersonation of the legitimate destination for the purposes of denial of service.

Candidate technology:

- OASIS SOAP Message Security
- MIME with XML Signature/XML Encryption

- XML Signature as used apart from OASIS SOAP Message Security and SOAP message exchanges, e.g. for identification and authentication of payloads

Threat association:

T-03, T-04, T-05, T-06, T-07, T(OOS)-01, T(OOS)-03, T(OOS)-04, T(OOS)-08), T(OOS)-13, T(OOS)-14.

C-03: Data Integrity

Definition: Data integrity: The property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner (see [RFC 2828]).

Explanation: Data in a web services context is taken to mean a SOAP message or portions of a SOAP message, including one or more SOAP headers, a body, or attachment parts. Although data integrity is concerned with allowing a recipient of data to detect changes, whether accidental or malicious, data origin authentication mechanisms are required in conjunction with data integrity mechanisms in order to protect against active substitution and forgery attacks. When only providing integrity for portions of content, care must be taken to protect against subtle attacks, especially when a message is targeted at SOAP intermediaries as well as an ultimate receiver.

Note that the term “Integrity” is generally used differently in the field of information management to mean that the data is correct, proper, accurate, and consistent with other data or the real world. In this sense it usually implies that there are well-regulated procedures of creating, modifying and deleting the data. Here we are using “Integrity” in the security sense of not being altered without detection of such alteration even when under active attack.

Threat association: T-01. Additional threats associated with sub-categories of data integrity are listed below. Note that when used in conjunction with data origin authentication T-03, T-04 and T-05 are addressed.

3.2.1 C-03A: Transport Data Integrity

Definition:

Transport Data Integrity: Data integrity provided by the protocol layer that SOAP messages are bound to, e.g. HTTP secured by SSL/TLS (HTTPS).

Explanation: Transport integrity is applied to the entire SOAP message and may also include underlying protocol layers. For example, with HTTPS the HTTP message is also protected. Such transport layer security is “transient” in that the integrity is only effective while the transport session exists. Transport integrity is not appropriate for end-to-end security (from SOAP initiator to ultimate receiver) when SOAP intermediaries are present, since SOAP processing rules allow intermediaries to make changes to the SOAP message, and since transport protection is not in effect during intermediary processing.

Candidate technology:

- SSL/TLS with encryption enabled.

Additional Threat Associations: T-08, T(OOS)-10, T(OOS)-14.

3.2.2 C-03B: SOAP Message Integrity

Definition:

Soap Message Integrity**:** Data integrity applied at the SOAP Messaging layer in a manner that allows SOAP processing rules to be followed.

Explanation: SOAP message data integrity is for a web service message that may be processed by SOAP intermediaries and may exist for extended periods of time at intermediary and/or ultimate receiver SOAP nodes before being processed. The intention is to protect message data even when not in transit, such as before processing is completed. An example is a SOAP message waiting at a SOAP node for aggregation with other content yet to be processed. Transport integrity is inappropriate for such cases since it terminates with the transport session.

SOAP message integrity should be applied to a SOAP message in a manner that enables processing by SOAP intermediaries, which suggests that integrity protecting a combination of SOAP header blocks the body and attachments is preferable to protecting the entire SOAP envelope element or the entire SOAP header element. Protection may also include SOAP attachments.

Candidate technologies:

- XML Signatures as profiled in the OASIS SOAP Message Security specification. Note that keys may be conveyed out of band or with the message using a SOAP Message Security token profile, including (but not limited to) Username tokens (for derived keys) [UTP 1.0], X.509 [X509 1.0], Kerberos tokens, SAML tokens [SAML 1.0], REL tokens [REL 1.0], or others.
- XML Signatures with MIME, not in the context of SOAP Message Security (out of scope)

XML Signatures not in the context of SOAP Message Security headers can be used by applications, but that use is not addressed in this document.

3.3 C-04: Data Confidentiality

Definition: Data confidentiality: The property that information is not made available or disclosed to unauthorized individuals, entities, or processes [i.e. to any unauthorized system entity].

Explanation: The property that eavesdroppers or other unauthorized parties cannot view confidential message content. Typically this is achieved with encryption. Note that confidentiality is a distinct concept from privacy, so in the definition "disclosure" refers to the ability to view or eavesdrop the information when transferred or processed. Confidentiality techniques may be used as one aspect of maintaining privacy, however.

Threat Associations: T-02, T(OOS)-10, T(OOS)-14.

Disclosure related attacks as well as attacks that reduce the confidentiality strength (e.g. man-in-the-middle SSL/TLS cipher suite attacks) are relevant.

3.3.1 C-04A: Transport Data Confidentiality

Definition: Data confidentiality provided by the protocol layers that SOAP messages are bound to in a transport protocol stack specific manner. An example is HTTP secured by SSL/TLS (HTTPS).

Explanation: Data confidentiality is applied to the entirety of the SOAP message as well as possibly other protocol layers (e.g. HTTP when SSL/TLS is in use). With end-to-end confidentiality between the initial SOAP sender and the ultimate receiver this prevents the use of SOAP intermediaries.

Candidate technology:

- SSL/TLS with encryption enabled.

Additional threat associations:

none.

3.3.2 C–04B: SOAP message confidentiality

Definition: Data confidentiality applied at the SOAP messaging layer in a manner that allows SOAP processing rules to be followed.

Explanation: SOAP message confidentiality supports the confidentiality requirements unique to SOAP messaging, including:

1. SOAP intermediaries may be present and must be able to follow SOAP processing rules for the message, even when confidentiality has been applied.
2. Confidentiality may be applied to multiple portions of a SOAP message and be intended for different SOAP messaging participants.
3. A SOAP message (or portions) may retain confidentiality protection while not in transit.

   This may include extended periods of time that the SOAP message is queued at an intermediary or ultimate receiver before being processed. An example is a SOAP message waiting at a SOAP node for aggregation with other content yet to be processed.

Transport confidentiality is generally inappropriate for these requirements since it terminates with the transport session.

In order for SOAP message confidentiality to be applied to a SOAP message in a manner that enables processing by SOAP intermediaries, a combination of SOAP header blocks, body blocks and attachments is appropriate, but the soap:Envelope, soap:Header and soap:Body elements must be visible to all parties and should not be encrypted. The SOAP message must also remain well-formed XML.

Candidate technologies:

- XML Encryption, as profiled by the OASIS SOAP Message Security specification.

Additional threat associations: none

3.4 C-05: Message Uniqueness

Definition: the ability to insure that a specific message is not resubmitted for processing.

Explanation: Attacker could resend all or selective parts of a message causing undesirable side effects. For example, an attacker sending the same valid message moving money from one bank account to another bank account. The original message request is valid, but not its replay. Additionally, sending the same valid message is frequently used in many denial-of-service attacks. While an application solution against replay attacks may utilize message ordering and reliable message delivery mechanisms, this security challenge makes no attempts to address these issues.

Candidate technologies:

- At the transport layer, using SSL/TLS between the node generating the request and the node insuring for downstream nodes that this is a unique request.

- At the message layer, the sending and receiving SOAP nodes must do a combination of different things. The sender must sign SOAP message header nonce, creation time[, expiration time] and optional user data. This user data may include critical transactional information and service identification elements. The transactional data protects the actual user request. The optional service identification elements protect the replay of the signature to another service that utilizes the same message data. The receiving node must verify the signature and check that the creation time is not stale. Lastly, it must compare the received nonce with a cache of previously received nonces. This cache of nonces must be maintained until the associated expiration time or the creation time plus a hard-coded delta has expired. Note: when multiple servers are performing this functionality, some mechanism must be implemented to create a functional global cache across all these systems.

Threat association: T-07, T-08, T-09, T(OOS)-14.

4 Threats

This section details a list of traditional security threats. Note that in many cases the threats overlap. That is particular attacks may represent threats in several categories.

ID	Name	Description
T-01	Message Alteration	The message information is altered by inserting, removing or otherwise modifying information created by the originator of the information and mistaken by the receiver as being the originator's intention. There is not necessarily a one to one correspondence between message information and the message bits due to canonicalization and related transformation mechanisms.
T-02	Confidentiality	Information within the message is viewable by unintended and unauthorized participants. (e.g. a credit card number is obtained).
T-03	Falsified Messages	Fake messages are constructed and sent to a receiver who believes them to have come from a party other than the sender. For example, Alice sends a message to Bob. Mal copies some (or all of) it and uses that in a message sent to Bob who believes this new action was initiated by Alice. This overlaps with T-01. The principle is that there is generally little value to saying a message has not been modified since it was sent unless we know who sent it.
T-04	Man in the Middle	A party poses as the other participant to the real sender and receiver in order to fool both participants (e.g. the attacker is able to downgrade the level of cryptography used to secure the message). The term "Man in the Middle" is applied to a wide variety of attacks that have little in common except for their topology. Potential designs have to be closely examined on a case-by-case basis for susceptibility to anything a third party might do.
T-05	Principal Spoofing	A message is sent which appears to be from another principal (e.g. Alice sends a message which appears as though it is from Bob). This is a variation on T-03.
T-06	Forged claims	A message is sent in which the security claims are forged in an effort to gain access to otherwise unauthorized information (e.g. A security token is used which wasn't really issued by the specified authority). The methods of attack and prevention here are essentially the same as T-01
T-07	Replay of Message Parts	A message is sent which includes portions of another message in an effort to gain access to otherwise unauthorized information or to cause the receiver to take some action(e.g. a security token from another message is added).Note that this is a variation on T-01. Like "Man in the Middle" this technique can be applied in a wide variety of situations. All designs must be carefully inspected from the perspective of what could an attacker do by replaying messages or parts of messages.

ID	Name	Description
T-08	Replay	A whole message is resent by an attacker
T-09	Denial of Service	Amplifier Attack: attacker does a small amount of work and forces system under attack to do a large amount of work. This is an important issue in design and perhaps merits profiling in some cases.

Table 1: Threats

Additional information on security threats can be found in the following titles:

- Stallings, William. *Cryptography and Network Security: Principles and Practice (3rd Edition)*, Prentice Hall 2002
- Fisch, Eric A and White, Gregory B. *Secure Computers and Networks: Analysis, Design, and Implementation*, CRC Press, 1999
- Kaufman, Charlie and Perman, Radia and Speciner, Mike. *Network Security: Private Communication in a Public World*, Prentice Hall, 2002
- Ford, Warwick and Baum, Michael S. *Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption (2nd Edition)*, Prentice Hall, 2000
- *Schneier, Bruce. Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition*. John Wiley & Sons. 1995

5 Security Solutions, Mechanisms and Countermeasures

In this section, we provide a high-level description of security solutions, which are defined in terms of security layers that address the SOAP message security challenges in section 3. We then define the specific security mechanisms and associated countermeasures that are addressed by the Security Profiles.

Mechanisms to address security challenges may be applied at different communication layers and possibly in combination. The primary concerns of this document are the SOAP and transport layers. Within the transport layer the focus is primarily on HTTP and HTTPS. Combinations of security mechanisms in the layers may be applied to satisfy different security requirements.

SOAP layer mechanisms may be used to provide security for attachments.

This document focuses on scenarios for transport and SOAP layer security. Users may implement their own data (payload) layer security, but data layer security is not addressed explicitly in this document.

Transport and SOAP security layers can be configured to address a variety of security requirements. These variations are enumerated later in this section. We define abstract security functions that may be used to address the various security threats that we previously described in section 4.

5.1 Transport Layer Security Descriptions

The protocol layers that provide transport for the SOAP Messaging protocol (transport layer) may be used to provide security services to meet application or SOAP Messaging security requirements. This may be done in combination with SOAP message security mechanisms or independently. This section focuses on the transport mechanisms only. These mechanisms provide integrity and/or confidentiality for HTTP messages.

Because the only transport mechanism within the scope of this document is HTTP (optionally over SSL/TLS) we assume that each SOAP node has an associated HTTP node, which might be a part of the SOAP node or might be a distinct entity. We also assume that SOAP messages between nodes are carried on HTTP messages between their associated HTTP nodes. Communication between a SOAP node and its associated HTTP node is regarded as internal to a platform and we make no assumptions about its nature or the information transferred other than

- The SOAP message itself is communicated.
- When an HTTP request containing a SOAP message is sent over a connection that was established using some HTTP authentication mechanism, the HTTP server will communicate to its associated SOAP node the identity that was established by that authentication mechanism. We do not assume that it communicates any credential used to establish that identity.

Note in particular that we do not assume any communication between the associated HTTP and SOAP nodes with regards to the certificates used to establish a TLS/SSL connection.

In what follows when a word or phrase such as “N” refers to a specific SOAP node we use the notation “N-HTTP” to refer to its associated HTTP node.

5.1.1 Integrity

Integrity may be provided for an entire SOAP message using the transport layer. When SSL/TLS is used in conjunction with HTTP (HTTPS), the entire HTTP message, including the start-line (e.g. POST), HTTP headers, and body receives integrity protection. This SOAP message conveyed in the HTTP body is also protected. This integrity is only in effect for the duration of the HTTP session and provides no protection for SOAP messages once received (and possibly queued by the web service consumer or provider). Note that integrity is provided for the entire SOAP message – partial integrity is not possible with this mechanism. This mechanism is not suitable for end-end SOAP message integrity in the presence of SOAP intermediaries.

The basic operation of this mechanism is as follows:

1. SOAP node A's associated HTTP node initiates an HTTPS connection to another SOAP node B's associated HTTP node.
2. SSL/TLS session is established, starting integrity protection
3. SOAP messages are conveyed from A to B, potentially a SOAP message or fault is conveyed in the HTTP response
4. HTTP and SSL/TLS session is terminated, ending integrity protection

Note that the quality of SSL/TLS integrity protection depends on an adequate SSL/TLS cipher suite and key length being selected. Care must be taken in selection of cipher suites and key lengths to prevent downgrade attacks. Options with inadequate security should not be offered even if they are supported in the code. Determination of adequate levels of security is, of course, a matter of individual policy. However, the Profile will make some recommendations where appropriate.

5.1.2 Confidentiality

Confidentiality may be provided for an entire SOAP message using the transport layer. When SSL/TLS is used in conjunction with HTTP (HTTPS), the entire HTTP message including HTTP headers is protected as well. This confidentiality is only in effect for the duration of the HTTP session and provides no protection for SOAP messages once received (and possibly queued by the web service consumer or requestor). Confidentiality is applied to the entire SOAP message; partial confidentiality is not possible, making this unsuitable for SOAP messages to be conveyed through SOAP topologies involving SOAP intermediaries.

The basic operation of this mechanism is the same as that using transport layer to provide integrity. [Section 5.1.1

Note that the presence and quality of SSL/TLS integrity protection depends on an adequate SSL/TLS cipher suite and key length being selected. Care must be taken in selection of cipher suites and key lengths to prevent downgrade attacks. Options with inadequate security should not be offered even if they are supported in the code.

5.1.3 Authentication by HTTP Service

A SOAP node A whose associated HTTP node initiates a connection from SOAP node B's associated HTTP node may authenticate B using transport layer mechanisms such as SSL/TLS. In the SSL/TLS case the authentication consists of a server X.509 certificate combined with a proof of private key possession as part of the SSL/TLS protocol. In addition, some clients may perform additional checks such as comparing the service URL domain name against the certificate distinguished name, for example, to attempt to detect certificate substitution attacks. Finally, relying parties should perform a certificate validation check to ensure that the certificate was not revoked, either due to private key compromise or other reasons before relying on the validity of the authentication information.

The basic operation of the mechanism is as follows:

1. HTTP node associated with A initiates HTTPS connection to HTTP node associated with B.
2. As part of establishing SSL/TLS session, B's HTTP node authenticates to A's HTTP node
3. SOAP messages are conveyed from A to B, potentially SOAP message or fault is conveyed in HTTP response
4. HTTP and SSL/TLS session is terminated

Note that the authentication is for the session and that by default there is no lasting record or association of the authentication action with the SOAP message.

5.1.4 Authentication by HTTP User Agent

A SOAP node A whose associated HTTP node initiates a connection to SOAP node B's associated HTTP node may authenticate to SOAP node B. If B's HTTP node also authenticates to A's HTTP node it is said to be mutual authentication.

Note that a web service provider might authenticate at the transport layer and the web service consumer at the SOAP messaging layer, depending on the desired authentication properties.

An HTTP user agent authentication may be:

- HTTPS client X.509 certificate authentication,
- HTTP basic or digest authentication with HTTPS confidentiality
- HTTP basic or digest authentication without HTTPS confidentiality

5.1.4.1 HTTPS X.509 client Authentication

1. A's HTTP node initiates HTTPS connection to B's HTTP node
2. As part of establishing SSL/TLS session, web service consumer authenticates to provider using X.509 client certificate with private key proof of possession as part of SSL/TLS protocol
3. Once HTTPS session is established A sends SOAP messages and the HTTP response may convey a SOAP message or Fault.
4. HTTPS session is closed, ending authenticated transfer

5.1.4.2 HTTP Basic or Digest authentication with HTTPS Confidentiality

HTTP Basic and Digest authentication mechanisms are outlined in [RFC 2617],

1. A-HTTP node initiates HTTPS connection to B-HTTP node with HTTPS confidentiality (requires appropriate cipher suite etc)
2. HTTP Basic or Digest authentication performed as part of SOAP message request POST

HTTPS session is closed

Note that B-HTTP must request authentication explicitly. The SOAP message may be POSTed twice – once in the original POST that results in an HTTP response requesting authentication and then in the request that conveys the authentication information in the header. This could be an issue for large SOAP messages.

Adequate protection against replay attacks is required with HTTP authentication and POSTs as noted by RFC 2617. HTTPS confidentiality requires appropriate cipher suites and protection against downgrade attacks.

Using HTTP with Digest authentication provides no real benefits in terms of authentication over Basic authentication, although with the proper cipher suites it can provide integrity.

5.1.4.3 HTTP Basic or Digest Authentication in the clear

HTTP Basic or Digest authentication performed as part of HTTP session that includes SOAP message request POST.

Despite the risk of insider attack (most attacks are insider attacks) HTTP authentication without HTTPS may be appropriate within an enterprise or other secured environments. Protection against replay attacks is required as noted by RFC 2617.

5.1.5 Attributes

Attributes may be conveyed in HTTP header fields [RFC 2616]. This may require integrity and/or confidentiality protection using HTTPS, depending on application requirements.

Attributes may also be conveyed in the HTTPS client X.509v3 certificate through the use of certificate extensions, although this may not be interoperable. See PKIX RFC 3280.

5.1.6 Combinations

The preceding transport layer security mechanisms may be combined with each other as needed. The following table attempts to identify the combinations that we believe are significant with a unique tag that we will use in later sections.

Challenge Supported	Transport Layer Technologies being Utilized		Tag[2]	Comment
Integrity	SSL/TLS		BISP1	
Confidentiality	SSL/TLS			Assuming that cipher suites NULL-SHA or NULL-MD5 are not being supported because these suites do support encryption.
Provider (server) Authentication	SSL/TLS			Assume X.509 certificates being used to identify consumer and provider with mapping to trusted root CA.
Consumer (client) Authentication	SSL/TLS[3] with client authentication		BC1	
	HTTP Basic		BC2	
	HTTP Digest		BC3	
	HTTP Attributes		BC4	
	SSL/TLS	HTTP Basic	BC5	This assumes that BISP1 is also supported. Additionally, assumes cipher suites NULL-SHA & NULL-MD5 not supported, i.e., protection against downgrade attacks.
		HTTP Digest		

Table 2: Transport Level Security Options

The intention is for an application developer to select one or more solutions that address the relevant security challenges. For example, if consumer authentication is required then any one of the BCx solutions would meet this need.

As indicated, a single solution may meet multiple security challenges. For example, assuming cipher suites NULL-SHA or NULL-MD5 are not supported, using SSL/TLS will ensure transport layer integrity, confidentiality and provider authentication.

5.2 SOAP Message Layer Security Descriptions

Security services may be provided at the SOAP Messaging protocol layer using the SOAP Message Security specification from the OASIS SOAP Message Security technical committee in conjunction with token specifications developed in that committee. These security mechanisms may be combined with the transport layer security mechanisms discussed above.

[2] The tag naming convention consists of three parts. The first character is a “B” in the first character to identify that this is a binding level solution. (Note: “T” was not used because of possible confusion with “T” used by Threat tags.) The next 1 to 3 letters identify the transport challenge: “I” for Integrity, “S” for confidentiality (Secret), “P” for Provider authentication, and “C” for Consumer authentication. The last component is a number identifying the solution instance.

[3] Note: user can support NULL-SHA or NULL-MD5 cipher suites for this usage.

5.2.1 Integrity

Integrity may be provided to a portion or combination of SOAP message content using XML Digital Signature as outlined in the SOAP Message Security specification. Such integrity has the advantage that it remains with the SOAP message beyond an HTTPS session, suitable for providing end-end integrity despite SOAP intermediaries, when used properly.

1. SOAP Sender (either initial SOAP Sender or SOAP Intermediary) protects integrity of some portion or combination of SOAP body, attachments and header blocks using an XML Digital Signature placed in a wsse:Security header block targeted at the SOAP receiver relying on integrity. SOAP Sender may also convey key information using security tokens in the message header enabling relying party to verify signatures. Note that in some cases integrity may be relied upon by more than one SOAP receiver. In case portions of the message are persisted with their signature integrity may be relied upon by participants besides SOAP receivers.
2. Message is sent, potentially through one or more SOAP intermediaries. SOAP role associated with SOAP security header for integrity protection determines relying party. Depending on how SOAP role is defined integrity may be verified by multiple SOAP receivers.

5.2.2 Confidentiality

Confidentiality may be provided to portions or some number of SOAP Message content using XML Encryption as outlined in the SOAP Message Security specification. Note that encryption must not be applied so that SOAP message processing cannot be performed. SOAP message confidentiality protection has the advantage that it remains with the SOAP message beyond an HTTPS session, and is suitable for providing end-end confidentiality despite SOAP intermediaries when used properly.

1. SOAP Sender (either initial SOAP Sender or SOAP Intermediary) protects confidentiality of some combination of SOAP body, or header blocks or portions using XML Encryption as outlined in SOAP Message Security. Sender may also convey key information using security tokens in the message header.
2. Message is sent, potentially through one or more SOAP intermediaries. Depending on processing roles and rules, confidentiality may be applicable for one or more SOAP receivers. Special consideration must be given to either the replacement of encrypted data with clear data by intermediaries since this modification could break any signatures that referenced the encrypted data.

5.2.3 SOAP Sender Authentication

A SOAP Sender (either an initial SOAP sender or a SOAP intermediary) may provide authentication for one or more SOAP receivers by including one or more appropriate SOAP Message security tokens in security headers targeted at the receiver roles may be used in combination with XML Signatures as profiled by SOAP Message Security to provide confirmation of the token claims and to bind the claims to the message.

Note that in a SOAP message from a web service consumer to a web service provider, SOAP sender authentication authenticates the consumer. In a SOAP message from a web service provider to a web service consumer (such as conveyed in an HTTP response in a request-response MEP) then SOAP sender authentication authenticates the provider to the consumer. SOAP receiver authentication as such does not make sense given a one-way message.

5.2.4 Attributes

Attributes may be conveyed in application specific SOAP Message Security XML or Binary security tokens (SOAP Message Security extension points), or SOAP Message Security SAML Tokens conveying attribute assertions to give two examples.

5.2.5 Message Uniqueness

This functionality is build upon the message integrity mechanisms, digital signatures, referred to in Section 5.2.1 being applied to several fields with special semantics and a number of things outside the actual message exchange. Depending upon the type of security token being utilized by the application to authenticate the sender, different elements in the message may be utilized. All the solutions are built upon the following key types of information being present in the sender message:

Unique message identifier: this element is used to uniquely identify the message. No two messages should ever have this value. While this data could be consequently assigned sequence numbers or non-random data, experience has shown that such practices allow for session hijacking unless the associated authentication mechanisms are very strong. Using true random values for the message identifier is best practice because an attacker can not effectively guess what message identifier someone is using or may use. [Some form of this element must be present in any solution]

Timestamp: a time that bounds the associated message identifier lifetime. Without this value, the consuming entity would potentially have to maintain data to track all message identifiers that it has ever processed. For some restrictive environments, e.g., single source, this timestamp can be used for the unique message identifier. In general, this is not true. The bigger issue with the timestamp is that the sending and receiving systems must be loosely time synchronized so that the receiving system does not have to maintain an ever-increasing database of processed message identifiers. With the availability of clock synchronization protocols and the receiver ability to control the size of the time window, applications can control the degree of time synchronization needed. While careful date/time set up could work if an application supports a large time window, e.g., 5-10 minutes, in general some form of clock synchronization is really required for effective operation. [Some form of this element must be present in any solution]

Optional Application Restrictions: These elements allow an application to prevent the replay of the preceding elements to different receiving systems. For example, to prevent a valid message identifier and application message data from being sent to a different receiving system and being processed, the domain of the target service that this request is intended for could be included within the data to be signed. [This is application dependent data with associate application semantic checking.]

Of the different types of security tokens that our profile is committed to address, i.e., X.509 certificates, username, Kerberos, SAML and REL, only username tokens currently have elements defined that map to the unique message identifier and timestamp element just described.

As will become apparent, no security token profile nor other standard will deliver a fully operational solution to the message uniqueness challenge at the SOAP message layer.

5.2.5.1 Username Token

In particular, the username token profile defines the following elements that the sending system must populate when building a message uniqueness solution:

Nonce: a random value that the sender generates and uses as the unique message identifier. [The nonce is a recommended element in OASIS Username Token Profile that can be overloaded to serve as the unique message identifier. When used for replay prevention, this element must be present. When used for this purpose, it must be large enough to ensure that multiple simultaneous requesters do not generate the same nonce value causing a false positive.]

Creation Time: the time that the associated nonce was created. [The creation time is a recommended element in OASIS Username Token Profile that can be overloaded to serve as the timestamp. When used for replay prevention, this element or expiration time element must be present.]

Expiration Time: the time when the associated nonce is no longer valid to be used. [The expiration time is an optional element in OASIS Username Token Profile that can be overloaded to serve as the timestamp. If not present, then the receiving system must add an internally configured delta time to the creation time element.]

Additionally, the preceding required and optional data along with the username must be signed by the sender so that the receiving system can ensure that none of the preceding elements has been modified by an attacker. This comes with the unstated assumption that the signing key (some function of the associated password) is known only to the sender and receiver as either an out-of-band shared secret or encrypted. Otherwise, the receiver can not authenticate the sender is who then say they are.

On the receiving system, the receiver must perform the following actions:

1. Verifying the signature containing the nonce, timestamps and optional restriction data. Note: this check is completely independent from any other integrity checking that the sender/receiver may be performing.

2. Check that the expiration time (or creation time + maximum delta) is less than the current time.

3. Looking up the nonce value in a nonce cache. If the nonce value is already present, then fail the request. If the nonce value is not present, then add the nonce and expiration time values to the cache. If multiple receiving systems are concurrently active, then the nonce cache must be across all servers in the pool. Independently, the nonce cache should automatically delete expired nonces. Our intention is to describe the abstract processing that the receiver is performing, not the implementation specifics. [This functionality is application specific because no existing standard/protocol covers this functionality.]

4. Perform any application specific restriction checks, e.g., checking target domain. [This functionality is application specific because no existing standard/protocol covers this functionality.]

5.2.5.2 X.509 Certificate, Kerberos, SAML and REL Tokens

The OASIS X.509 Certificate, SAML and REL Token Profiles, as well as the upcoming OASIS Kerberos Token Profile, do not have the required elements that can act as a message identifier. This requires the application developer to define proprietary elements to address these needs outside of the scope of these token profiles.

5.2.5.3 Other Token Types

There are other token types being worked on that contain nonce and timestamp elements. However, their detail characteristics may prohibit them for being used to prevent replay attacks.

5.2.6 Combinations

The preceding message layer security mechanisms may be combined with each other as needed. The following table attempts to identify the combinations that we believe are significant with a unique tag that we will use in later sections.

Challenge Supported	Message Layer Technologies being Utilized		Tag[4]	Comment
Integrity	XML Digital Signature		SI1	
Confidentiality	XML Encryption		SC1	
SOAP Sender Authentication	XML Encryption	username & [password|digest]	SA1	Without the ability to encrypt password/ digest, sender open to man-in-middle stealing password/digest and reusing it.
	username & [password|digest]		SA2	SOAP Attributes
	X.509 Certificate		SA3	
	Kerberos Token[5]		SA4	
	SAML Token		SA5	
	REL Token		SA6	

Table 3: SOAP Message Level Security Options

The intention is for an application developer to select one or more solutions that address the relevant security challenges. For example, if SOAP sender authentication is required then any one of the SAx solutions would meet this need.

Missing from this table is SOAP receiver authentication. Receiver message layer authentication can only be supported by a response message in which the role of the sender and receiver has been exchanged, i.e., the sender is the provider.

5.3 Combining Transport Layer and SOAP Message Layer Mechanisms

As noted above security services may be provided at either or both the transport layer and the SOAP message layer. The choice often depends on application requirements, based on answers to questions such as:

1. Is it necessary to apply integrity and/or confidentiality at a granularity other than the entire SOAP message? This is usually true when SOAP intermediary processing is expected.
2. Does the protection need to exist beyond the transport session, protecting SOAP messages when queued at a SOAP node for example?

[4] The tag naming convention consists of three parts. The first character is a “S” in the first character to identify that this is a SOAP message level solution. The next letter identifies the type of SOAP message level challenge: “I” for Integrity, “C” for Confidentiality, “A” for SOAP sender Authentication. The last component is a number identifying the solution instance.

[5] Kerberos tokens are part of our charter candidate technologies. However, usage of this technology in this profile will be deferred until OASIS TC delivers this core specification. Note also that as other types of security tokens are added to our list of charter technologies, they will be added to these security profiles.

3. Is there a need to save evidence such as authentication assertions for subsequent dispute resolution?
4. Is there a need for transport layer protocol independence?
5. How important is interoperability of attribute information?

Special cases are noted in the sections above where additional mechanisms are required to ensure security. In general, minimizing combinations while following recommended security practices for the security technologies should reduce risks.

5.4 Transport and Message Layer Security Combinations

This section describes a selected subset of common security scenarios and identifies potential solutions for various security requirements. The security requirements vary from simple to complex depending upon the mechanisms selected and the underlying need. This approach allows the users to select a specific security scenario and implementation mechanisms that best meet their needs.

There are three basic categories of implementation solutions:

- transport layer,
- SOAP message layer
- hybrid that combines mechanisms from transport and SOAP message layers.

Figure 1 attempts to depict the potential solution space. It is organized with transport only mechanism on the left side of the figure and SOAP message mechanisms on the right side. Hybrid solutions occupy the space in the middle. This figure is not bound to any specific scenario. Different scenarios may be able to only support a subset of implementations, e.g., one-way scenario can not support SOAP mutual authentication because there is no SOAP response message.

Additionally, Figure 1 is organized from top to bottom to go from no security to increasing complex security solutions.

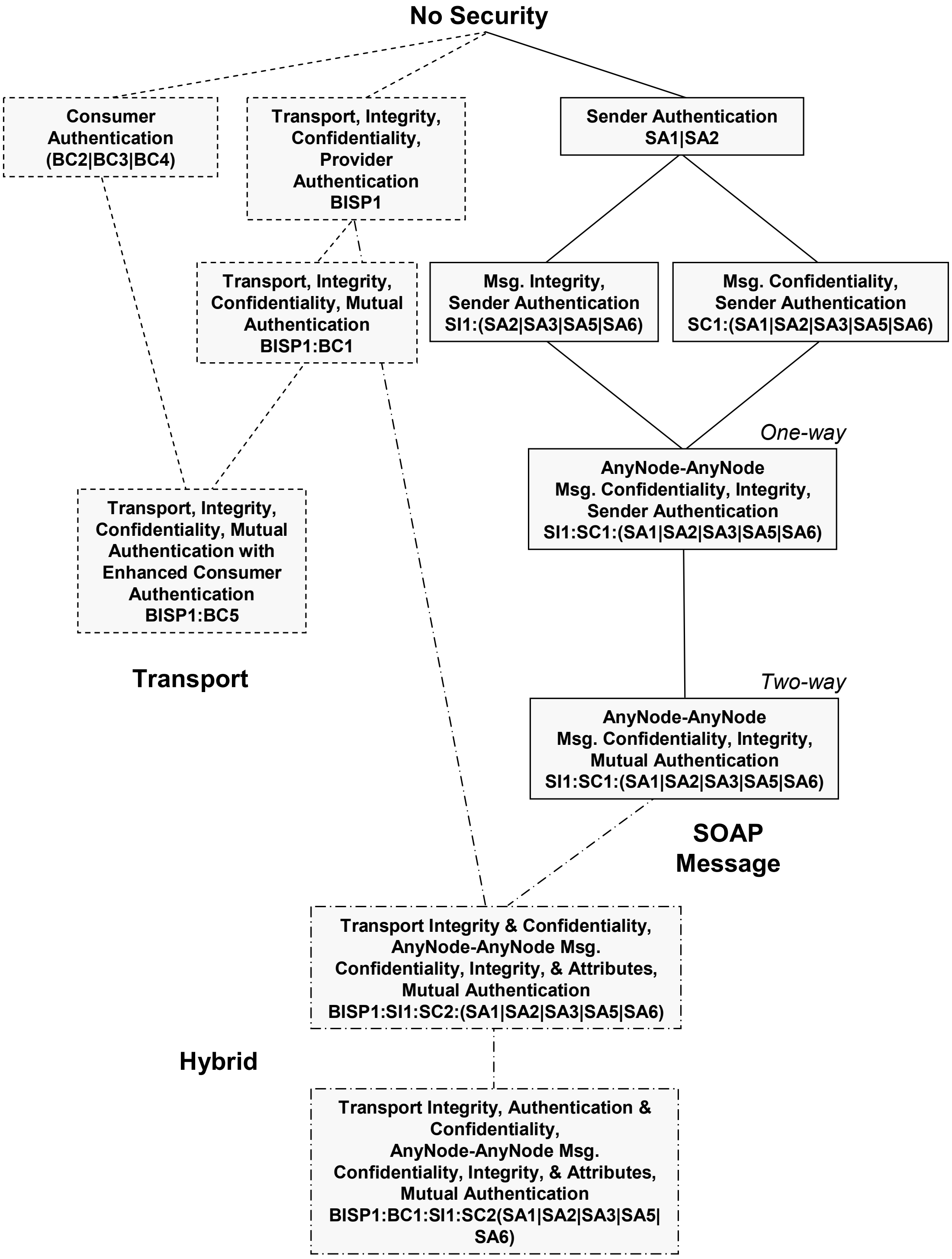
No Security
Consumer Authentication (BC2|BC3|BC4)
Transport, Integrity, Confidentiality, Provider Authentication BISP1
Sender Authentication SA1|SA2
Transport, Integrity, Confidentiality, Mutual Authentication BISP1:BC1
Msg. Integrity, Sender Authentication SI1:(SA2|SA3|SA5|SA6)
Msg. Confidentiality, Sender Authentication SC1:(SA1|SA2|SA3|SA5|SA6)
One-way
AnyNode-AnyNode Msg. Confidentiality, Integrity, Sender Authentication SI1:SC1:(SA1|SA2|SA3|SA5|SA6)
Transport, Integrity, Confidentiality, Mutual Authentication with Enhanced Consumer Authentication BISP1:BC5
Transport
Two-way
AnyNode-AnyNode Msg. Confidentiality, Integrity, Mutual Authentication SI1:SC1:(SA1|SA2|SA3|SA5|SA6)
SOAP Message
Transport Integrity & Confidentiality, AnyNode-AnyNode Msg. Confidentiality, Integrity, & Attributes, Mutual Authentication BISP1:SI1:SC2:(SA1|SA2|SA3|SA5|SA6)
Hybrid
Transport Integrity, Authentication & Confidentiality, AnyNode-AnyNode Msg. Confidentiality, Integrity, & Attributes, Mutual Authentication BISP1:BC1:SI1:SC2(SA1|SA2|SA3|SA5|SA6)

Figure 1 Common Security Solutions Hierarchy

The eleven solutions identified in Figure 1are a much smaller set than all possibilities of combined security solutions suggested by Table 2 on page 20 and Table 3 on page 25. A basic question is what approach or reasoning was used to reduce the numbers? Starting with the four transport entries, the two left solutions: BISP1 and BISP1:BC1, are simply SSL/TLS with and without client authentication. The BC2 | BC3 | BC4 solution is all that can be done with only using HTTP. The last solution is simply the merging/ enhancement of the SSL/TLS solutions and the pure HTTP solution. Remember that these two transport level mechanisms: HTTP and SSL/TLS, only work between HTTP/TCP level nodes. No SOAP intermediaries are allowed. If multiple HTTP or higher nodes are encountered, then multiple instances of the transport layer mechanisms between all communication HTTP nodes may need to be used. Additionally, each intermediary has full access to all of the data passing by to look at or alter, i.e., no way to insure the integrity or confidentiality within the HTTP/TCP intermediaries.

Moving to pure SOAP message solutions, the top solution is identification of the sender, without integrity or confidentiality. The next two solutions are message level integrity or confidentiality along with the identification of who the sender (signer/encryptor) is. The assumption is that usually it does not matter if a message is unchanged unless you know who signed (originated) the data. Similarly, the secrecy of a message is not important if you can not also insure that source of the secret information. The two SI1:SC1:(SA1|SA2|SA3|SA5|SA6) solutions utilize all the SOAP message level mechanisms: Integrity, Confidentiality and Sender Authentication, for one-way and two-way MEP, respectively. Unlike the transport level mechanisms, the SOAP message level mechanisms allow integrity, confidentiality and sender authentication of all or part of a message to occur between any SOAP nodes, not just the ultimate sender and receiver.

Lastly, there is a pair of hybrid cases supported. The first hybrid case uses SSL/TLS to insure the confidentiality and integrity of the entire SOAP message data. The usage of SSL/TLS is a simple solution that also protects against various types of man-in-the-middle replay attacks that would be more complex and expensive to protect against via pure SOAP message level mechanisms. The bottom line is that this solution allows stricter security requirements to be imposed between a single pair of sender and receiver HTTP/TCP nodes than between other nodes in the message exchange. This is just the logical extension that each set of nodes in a complex message exchange may have different security requirements. Transport level mechanisms address only security requirements between connected HTTP/TCP nodes, while SOAP message level mechanisms addresses security requirements between any nodes in a message exchange. Each mechanism can be used multiple times for each combination of nodes that has specific security needs. The second hybrid case is identical to the first, but adds transport-level, mutual authentication of HTTP nodes to the scenario.

5.5 Security Considerations for Combinations

In this section we provide an overview of the issues to consider when deploying the combinations of transport and message layer security mechanisms defined in Section 5.4. For each of the common security solutions previously shown in Figure 1, we summarize the properties of the solution, threats addressed, and limitations.

These considerations may be used as a guide to select an appropriate security solution for many Web Services application deployments. By matching up a particular application’s security requirements against the solutions in this list, it should be possible in most cases to select an optimal combination of transport and/or message layer security mechanisms for that application.

5.5.1 Transport Layer Security Solutions

The solutions in this subsection are based solely on transport layer security mechanisms.

5.5.1.1 Consumer Authentication – BC2|BC3|BC4

This solution has the following properties:

- Provides authentication of the initial SOAP sender (or prior Intermediary) HTTP Node to the ultimate SOAP receiver (or latter Intermediary) HTTP Node when they are on adjacent HTTP Nodes.

5.5.1.1.1 Threats addressed

T-05

5.5.1.1.2 Limitations

- Is only appropriate between adjacent HTTP Nodes not from initial Sender to the ultimate Receiver when there are intermediaries.
- Does not provide authentication of the ultimate SOAP receiver (or latter Intermediary) HTTP Node to the initial SOAP sender (or prior Intermediary) HTTP Node.
- Does not provide origin authentication for the SOAP message (only provides authentication of the HTTP Node).
- Does not provide integrity of a SOAP message.
- Does not provide confidentiality of a SOAP message.
- Does not provide detection of replay of a SOAP message.
- Does not address Man in the Middle principal spoofing attacks.

5.5.1.2 Transport Integrity, Confidentiality, Provider Authentication – BISP1

This solution has the following properties:

- Provides integrity protection for a SOAP message while in transit from HTTP node to HTTP node.
- Provides confidentiality protection for a SOAP message while in transit from HTTP node to HTTP node.
- Provides authentication of the ultimate SOAP receiver (or latter Intermediary) HTTP Node to the Initial SOAP sender (or prior Intermediary) HTTP Node when they are on adjacent HTTP Nodes.

5.5.1.2.1 Threats addressed

T-01, T-02

5.5.1.2.2 Limitations

- Is only appropriate between adjacent HTTP Nodes.
- Does not provide authentication of the Initial SOAP sender (or prior Intermediary) HTTP Node to the ultimate SOAP receiver (or latter Intermediary) HTTP Node.
- Does not provide origin authentication for the SOAP message (only provides authentication of the HTTP Node).
- Does not provide detection of replay of a SOAP message.

5.5.1.3 Transport Integrity, Confidentiality, Mutual Authentication – BISP1:BC1

This solution has the following properties:

- Provides integrity protection for a SOAP message while in transit from HTTP node to HTTP node.
- Provides confidentiality protection for a SOAP message while in transit from HTTP node to HTTP node.
- Provides authentication of the ultimate SOAP receiver (or latter Intermediary) HTTP Node to the Initial SOAP sender (or prior Intermediary) HTTP Node when they are on adjacent HTTP Nodes.
- Provides authentication of the Initial SOAP sender (or prior Intermediary) HTTP Node to the ultimate SOAP receiver (or latter Intermediary) HTTP Node when they are on adjacent HTTP Nodes.

5.5.1.3.1 Threats addressed

T-01, T-02, T-03, T-04, T-05, T-06, T-07, T-08

5.5.1.3.2 Limitations

- Is only appropriate between adjacent HTTP Nodes.
- Does not provide origin authentication for the SOAP message (only provides authentication of the HTTP Node).

5.5.1.4 Transport Integrity, Confidentiality, Mutual Authentication with Enhanced Consumer Authentication – BISP1:BC5

This solution has the following properties:

- Provides integrity protection for a SOAP message while in transit from HTTP node to HTTP node.
- Provides confidentiality protection for a SOAP message while in transit from HTTP node to HTTP node.
- Provides authentication of the ultimate SOAP receiver (or latter Intermediary) HTTP Node to the Initial SOAP sender (or prior Intermediary) HTTP Node when they are on adjacent HTTP Nodes.
- Provides authentication of the Initial SOAP sender (or prior Intermediary) HTTP Node to the ultimate SOAP receiver (or latter Intermediary) HTTP Node when they are on adjacent HTTP Nodes.

5.5.1.4.1 Threats addressed

T-01, T-02, T-03, T-05, T-06, T-07, T-08

5.5.1.4.2 Limitations

- Is only appropriate between adjacent HTTP Nodes.
- Does not provide origin authentication for the SOAP message (only provides authentication of the HTTP Node).
- Does not address Man in the Middle principal spoofing attacks.

5.5.2 SOAP Message Layer Security Solutions

The solutions in this subsection are based solely on SOAP message layer security mechanisms.

5.5.2.1 Sender Authentication – SA1|SA2

This solution has the following properties:

- Provides sender authentication of SOAP message.

5.5.2.1.1 Threats addressed

T-05

5.5.2.1.2 Limitations

- Does not provide confidentiality of a SOAP message
- Does not provide integrity of a SOAP message.
- Does not provide origin authentication of a SOAP message.
- Does not provide detection of replay of a SOAP message.
- Does not provide authentication of HTTP nodes.
- Does not address Man in the Middle principal spoofing attacks.

5.5.2.2 Message Integrity, Sender Authentication – SI1:(SA2|SA3|SA5|SA6)

This solution has the following properties:

- Provides sender authentication of SOAP message.
- Provides end-to-end integrity protection for a SOAP message.
- Provides origin authentication of a SOAP message.

5.5.2.2.1 Threats addressed

T-01, T-05

5.5.2.2.2 Limitations

- Does not provide confidentiality of a SOAP message.
- Does not provide authentication of HTTP Nodes.
- Does not provide detection of replay of a SOAP message.

5.5.2.3 Message Confidentiality, Sender Authentication – SC1:(SA1|SA2|SA3|SA5|SA6)

This solution has the following properties:

- Provides end-to-end confidentiality protection for a SOAP message.
- Provides sender authentication of SOAP message.

5.5.2.3.1 Threats addressed

T-02, T-05

5.5.2.3.2 Limitations

- Does not provide integrity of a SOAP message.

- Does not provide authentication of HTTP Nodes.
- Does not provide detection of replay of a SOAP message.

5.5.2.4 One-Way AnyNode – AnyNode Message Confidentiality, Integrity, Sender Authentication – SI1:SC1:(SA1|SA2|SA3|SA5|SA6)

This solution has the following properties:

- Provides end-to-end integrity protection for a SOAP message.
- Provides end-to-end confidentiality protection for a SOAP message.
- Provides sender authentication of SOAP message.
- Provides origin authentication of a SOAP message.

5.5.2.4.1 Threats addressed

T-01, T-02, T-05, T-06

5.5.2.4.2 Limitations

- Does not provide authentication of HTTP Nodes.
- Does not provide detection of replay of a SOAP message.

5.5.2.5 Two-Way AnyNode – AnyNode Message Confidentiality, Integrity, Mutual Authentication – SI1:SC1:(SA1|SA2|SA3|SA5|SA6)

This solution has the following properties:

- Provides end-to-end integrity protection for a SOAP message.
- Provides end-to-end confidentiality protection for a SOAP message.
- Provides sender authentication (both consumer and provider) of SOAP message.
- Provides origin authentication of a SOAP message.

5.5.2.5.1 Threats addressed

T-01, T-02, T-05, T-06

5.5.2.5.2 Limitations

- Does not provide authentication of HTTP Nodes.
- Does not provide detection of replay of a SOAP message.

5.5.3 Hybrid Security Solutions

The solutions in this subsection are based on a combination of transport and SOAP message layer security mechanisms.

5.5.3.1 Transport Integrity and Confidentiality, AnyNode – AnyNode Message Confidentiality, Integrity, Mutual Authentication – BISP1:SI1:SC1:(SA1|SA2|SA3|SA5|SA6)

This solution has the following properties:

- Provides integrity protection for a SOAP message while in transit from HTTP node to HTTP node.

- Provides confidentiality protection for a SOAP message while in transit from HTTP node to HTTP node.
- Provides authentication of the ultimate SOAP receiver (or latter Intermediary) HTTP Node to the Initial SOAP sender (or prior Intermediary) HTTP Node when they are on adjacent HTTP Nodes.
- Provides end-to-end integrity protection for a SOAP message.
- Provides end-to-end confidentiality protection for a SOAP message across HTTP nodes.
- Provides sender authentication (both consumer and provider) of SOAP message.
- Provides origin authentication of a SOAP message.

5.5.3.1.1 Threats addressed

T-01, T-02, T-03, T-04, T-05, T-06, T-07, T-08

5.5.3.1.2 Limitations

- None

5.5.3.2 Transport Integrity and Confidentiality, Mutual Authentication, AnyNode – AnyNode Message Confidentiality, Integrity, Mutual Authentication – BISP1:BC1:SI1:SC1:(SA1|SA2|SA3|SA5|SA6)

This solution has the following properties:

- Provides integrity protection for a SOAP message while in transit from HTTP node to HTTP node.
- Provides confidentiality protection for a SOAP message while in transit from HTTP node to HTTP node.
- Provides authentication of the ultimate SOAP receiver (or latter Intermediary) HTTP Node to the Initial SOAP sender (or prior Intermediary) HTTP Node when they are on adjacent HTTP Nodes.
- Provides authentication of the Initial SOAP sender (or prior Intermediary) HTTP Node to the ultimate SOAP receiver (or latter Intermediary) HTTP Node when they are on adjacent HTTP Nodes.
- Provides end-to-end integrity protection for a SOAP message.
- Provides end-to-end confidentiality protection for a SOAP message across HTTP nodes.
- Provides sender authentication (both consumer and provider) of SOAP message.
- Provides origin authentication of a SOAP message.

5.5.3.2.1 Threats addressed

T-01, T-02, T-03, T-04, T-05, T-06, T-07, T-08

5.5.3.2.2 Limitations

- None

6 Scenarios

This section contains descriptions of scenarios, security requirements that might be imposed by applications using those scenarios and ways to satisfy those requirements (called solutions).

6.1 Notation for Describing Scenarios

The content of a scenario and the conventions used to describe them are as follows.

- An introductory paragraph in English
- SOAP nodes: A list of the SOAP nodes participating in the scenario. These are given arbitrary labels. Some of these labels may have been mentioned by name in the introductory paragraph. In describing a scenario with intermediaries it is sometimes convenient to give a single node two names. When that is done it will be noted with a notation such as

  $N_k = B$
- HTTP Sessions: A list of HTTP sessions that will carry messages. The notation

  $S: A \rightarrow B$

  Indicates A-HTTP is the HTTP User Agent that initiates session S talking to HTTP Service B-HTTP. Sessions might be created during the scenario or might have existed before the scenario begins.
- SOAP Messages: A SOAP message path that might include intermediaries carries a single SOAP message. Note that this means there is no specific content associated with a "SOAP Message" The notation

  $M: A \rightarrow B \rightarrow \dots \rightarrow Z$

  indicates that the scenario includes a SOAP message that travels on the indicated SOAP Path. Nodes in this description of a SOAP message are said to be prior to Nodes to their right and later than Nodes to their left in the SOAP message path.
- Hops: A Hop describes the transmission in an HTTP message of data related to a SOAP message. A Hop is not itself a SOAP message because in common usage "SOAP message" refers to a more abstract entity that includes all the hops on a SOAP message path.
  The notation

  H: A → B (Session S, Message M)

  indicates that H is an HTTP Message that is sent by A-HTTP to B-HTTP as part of transmission of SOAP message M. Nodes A and B are said to be adjacent (on Message M). Whether H is an HTTP request or response depends on whether A or B initiated HTTP Session S. If it is a response, the Hop to which it is a response will be indicated.

  H: A → B (Session S, Message M, Response to R)

  The order in which the Hops are listed is the order in which the HTTP messages are sent.
- Security Requirements: This section will contain any Security Requirements that are specific to this scenario and any modification of generic security requirements (as specified in section 6.4) that are required to make them applicable to this scenario.

6.2 Conventions for Describing Security Requirements and Solutions

The description of a security requirement contains:

- A short title for the requirement
- A description of a security related problem that might be solved using the technologies within our scope.
- A list of threats (from Section 4) that might subvert potential solutions
- A list of challenges (from Section 3) that the requirement participates in.
- A list of possible mechanisms called "solutions" that can be used to satisfy this requirement. Each solution can be qualified by conditions that must be satisfied for the solution to be applicable.

6.3 Terminology

In describing the scenarios, requirements and solutions, the following phrases are used.

- Node N supplies content X: N-HTTP is the HTTP Sender in a Hop whose HTTP Message contained some bytes interpreted in the SOAP Layer as X. If content is originally supplied on a Hop by SOAP node A, and SOAP Intermediary B then passes it on unchanged in a Hop to SOAP node C. That content is still regarded as having been supplied by SOAP node A.
- N-HTTP initiates an HTTP session: N-HTTP acting as an HTTP User Agent created a session by opening a connection to some HTTP Service associated with some other SOAP node.
- N-HTTP accepts an HTTP session: N-HTTP acting as an HTTP Service accepts an Http becomes a participant in an Http session by accepting an Http Request.

6.4 Generic Security Requirements

This section contains security requirements that may be imposed by applications that use the scenarios. The requirements in this section are generic to all scenarios and might apply to any uses of SOAP Messaging.

This section only presents security requirements for which solutions are available within the profiled technologies. Other security requirements that might exist must be addressed by application level mechanisms.

6.4.1 Requirement: Peer Authentication

A SOAP node A must be able to authenticate to any SOAP node B.

Threats: T-04, T-05

Challenges: C-01

Security solutions:

The following solution may be used to provide authentication of A to B when A is prior to B on a SOAP message Path.

a) SOAP Sender Authentication (Section 5.2.3) of the SOAP message.

The following solutions may only be used to provide authentication of A to B when A-HTTP initiates a session to B-HTTP.

b) HTTPS X.509 Client Authentication (Section 5.1.4.1

c) HTTP Basic or Digest Authentication with HTTPS Confidentiality (Reference 5.1.4.2)

d) HTTP Basic of Digest Authentication in the Clear (Reference 5.1.4.3)

The following solution may only be used to provide authentication of B to A when A-HTTP initiates a session to B-HTTP.

e) HTTPS X.509 Server Authentication (Section 5.1.4.1)

Solutions (c) and (d) do not address T-04 (man in the middle)

6.4.2 Requirement: Origin Authentication

A party A in possession of a party's (B's) public key must be able to prove that signed SOAP message content was produced by party A. And it must be possible to retain that ability as long as the SOAP message is retained.

Threats: T-04, T-05, T(OOS)-13

Challenges: C-01, C-05

Security solution:

a) Digital Signature on Message. SOAP Message Layer Integrity (Section 5.2.1)

6.4.3 Requirement: Integrity

A SOAP node B must be able to detect alteration of content supplied by a SOAP node A

Threats: T-01

Challenges: C-03

Security solution:

The following solution may be used to provide integrity for any content supplied by SOAP node A.

a) SOAP Layer Integrity (Section 5.2.1

The following solution may be used to provide integrity for any content while it is in transit on a Hop to or from A.

b) Transport Layer Integrity (Section 5.1.1

6.4.4 Requirement: Confidentiality

A SOAP node B must be able to exclusively access confidential content supplied by a SOAP node A and intended for SOAP node B.

Threats: T-02

Challenges: C-04

Security solution:

The following solution may be used to provide confidentiality of any content supplied by Node A

a) SOAP Layer Confidentiality (Section 5.2.2)

The following solution may be used to provide confidentiality for content while in transit from A-HTTP to B-HTTP

b) Transport Layer Confidentiality (Section 5.1.2)

6.4.5 Requirement: Message Uniqueness

A SOAP node B must be able to detect that a previous received message or part of a previous message from SOAP node A has been replayed.

Threats: T-07, T-08, T-09

Challenges: C-05

Security solution:

The following solution may be used to provide replay protection for any content received by SOAP node

a) Transport Layer Integrity (Section 5.1.1). Currently, there is no application interoperability solution at the SOAP message layer.

6.5 Scenario Descriptions

6.5.1 Scenario: One-Way

A SOAP message is sent over a SOAP message path from a SOAP node N_0 through zero or more SOAP Intermediaries to a SOAP node N_k using a series of HTTP Requests.

This scenario applies to situations where the loss of individual SOAP messages is insignificant (for example, in a status monitoring scenario where periodic status update events are provided such that if one update event is lost, a subsequent update event will convey correct status). No SOAP message response is generated by N_k or expected by N_0. Regardless of the protocol implemented by the transport layer, N_0 receives no SOAP message response.

The transport layer may not guarantee delivery of the SOAP message. The N_0 or any SOAP Intermediary may not be aware whether a SOAP message was successfully sent or delivered to, received or processed by, any other node. Receipt of an HTTP Response indicates that at the very least that the HTTP Node associated with the receiver has received the HTTP Request but does not guarantee that the SOAP message will ever arrive at the receiver.

SOAP Nodes:

- N_0
- [OPTIONAL] N_1, N_2, ... N_{k-1} (SOAP Intermediaries)
- N_k

HTTP Sessions:

- (for r=1,...,k-1) $S_r : N_r \rightarrow N_{r+1}$

SOAP Messages:

- M: $N_0 \rightarrow ... \rightarrow N_k$

Hops:

- (for r = 1, ... k –1) H_r: $N_r \rightarrow N_1$ (Session S_r)

Security Requirements

None beyond generic requirements of Section 6.4

6.5.2 Scenario: Synchronous Request/Response

This scenario is derived from the Synchronous Request/Response scenario in the WS-I Basic Applications Usage Scenarios [BPSA UsageScenarios]

A SOAP message (called the request) is sent from a SOAP node N_0 through zero or more SOAP Intermediaries to a SOAP node N_k. A SOAP message called the response is sent by N_k to N_0. The SOAP Path of this SOAP message is the reverse of that of the request. The Hops used in the transmission of the response are the HTTP responses to the Hops used in the transmission of the request.

SOAP Nodes:

- N_0
- [OPTIONAL] N_1, N_2, ... N_{k-1} (SOAP Intermediaries)
- N_k

Sessions:

- (for r = 0, ..., k-1) S_0: $N_0 \rightarrow N_1$

SOAP Messages:

- REQUEST: $N_0 \rightarrow N_1 \rightarrow ... N_k$
- RESPONSE: $N_k \rightarrow N_{k-1} \rightarrow ... N_0$

Hops:

- (for r = 0, ..., k-1) H-REQ$_r$: $N_r \rightarrow N_{r+1}$ (Session S_r, Message REQUEST)
- (for r = k, ..., 1) H-RESP$_r$: $N_r \rightarrow N_{r-1}$ (Session S_{r-1}, Message RESPONSE, response to H-REQ$_{r-1}$)

Security Requirements

None beyond generic requirements of Section 6.4

6.5.3 Basic Callback

This scenario was derived from the Basic call back scenario in the WS-I Basic Sample Applications Usage Scenarios. [BPSA UsageScenarios]

The first SOAP Message APPLICATION-REQUEST is sent from Node A through zero or more to Node B through a series of Hops. APPLICATION-REQUEST contains information that indicates where B should send the APPLICATION-RESPONSE.

B sends a SOAP Message (acknowledgement) to A through the Http responses of the same set of Hops

After APPLICATION REQUEST is processed B sends a SOAP Message APPLICATION-RESPONSE to A through zero or more intermediaries through a series of Hops.

A sends a SOAP Message (acknowledgement) to B through the HTTP response across the same set of Hops.

The APPLICATION-REQUEST and APPLICATION RESPONSE are related via correlation information that is provided by A in APPLICATION-REQUEST and duplicated by B into APPLICATION-RESPONSE.

SOAP Nodes:

- A = AP-REQ$_0$ = AP-RESP$_l$
- B = AP-REQ$_k$ = AP-RESP$_0$
- [OPTIONAL] AP-REQ$_1$, AP-REQ$_2$, ... AP-REQ$_{k-1}$ (SOAP Intermediaries)
- [OPTIONAL] AP-RESP$_1$, AP-RESP$_2$, ... AP-RESP$_{l-1}$ (SOAP Intermediaries)

Sessions:

- (for r = 0, ..., k-1) REQ-SESSION$_r$: AP-REQ$_r$ → AP-REQ$_{r+1}$
- (for r = 0, ..., l-1) RESP-SESSION$_r$: AP-RESP$_r$ → AP-RESP$_{r+1}$

SOAP Messages:

- APPLICATION REQUEST: A → AP-REQ$_1$ → ... → AP-REQ$_{k-1}$ → B
- ACK-1: B → AP-REQ$_1$→ ... → AP-REQ$_l$ →A
- APPLICATION RESPONSE: B →AP-RESP$_1$ → ... → AP-RESP$_{l-1}$ →A
- ACK-2: A → AP-RESP$_j$ → ... →AP-RESP$_1$ → B

Hops:

- (for r = 0, ..., k-1) REQ-HOP$_r$: AP-REQ$_r$ → AP-REQ$_{r+1}$
  (Session AP-REQ$_r$, Message APPLICATION REQUEST)
- (for r = k-1, ..., 0) ACK-1-HOP$_r$: AP-REQ$_{r+1}$ → AP-REQ$_r$
  (Session AP-REQ$_r$, Message ACK-1, Http response)
- (for r = 0, ..., l-1) RESP-HOP$_r$: AP-RESP$_r$ → AP-RESP$_{r+1}$
  (Session AP-RESP$_r$, Message APPLICATION RESPONSE)
- (for r = l-1, ..., 0) ACK-2-HOP$_r$: AP-RESP$_{r+1}$ → AP-RESP$_r$
  (Session AP-RESP$_r$, Message ACK-2, Http response)

Security Requirements:

Requirement: Message Correlation

SOAP Node A must be able to securely determine whether content of hop AP-RESP$_{r+1}$ supplied by SOAP Node B was generated in response to APPLICATION-REQUEST. This requirement addresses the fact that related messages may be delivered on unrelated sessions.

Threats: T-01, T-02, T-03, T-04, T-05, T-08, T-09

Challenges: C-01, C-02, C-03, C-04

Security solutions:

Providing a solution for this requirement would require composition of a solution using techniques that are not described in the documents that are in scope for this profile.

An example of a solution would be for SOAP Node A to provide (with confidentiality, integrity and authentication) some correlation information X along with the content C. SOAP Node B would provide (with confidentiality, integrity and authentication) the same correlation information X along with the application level response.

Requirement: Node Correlation

SOAP Node A must be able to securely determine whether the content of AP-RESP$_{r+1}$ was supplied by SOAP Node B in response to content C sent to SOAP Node B.

This requirement addresses the possibility that the credential Q used by SOAP Node A to identify SOAP Node B when targeting content to SOAP Node B is not the same credential R used by SOAP Node B to identify itself when targeting content to SOAP Node A.

Threats: T-01, T-02, T-03, T-04, T-05, T-08, T-09

Challenges: C-01, C-02, C-03, C-04

Security solution:

Providing a solution for this requirement would require composition of a solution using techniques that are not described in the documents that are in scope for this profile.

The simplest example of a solution, based on the example given for Message Correlation, would be to ensure that the same credential was used to provide confidentiality to, and authentication from, SOAP Node B (Q = R). A more complex solution, still based on the Message Correlation example, would require SOAP Node A to have access to some mapping of several credentials to SOAP Node B (Q => B and R => B).

7 Out of Scope

This section contains discussions of security aspects that are not considered in the security requirements of the scenarios. It is included so that the reader is aware that these have not been overlooked. The primary reasons that they are not considered is that mechanisms to deal with them are not present within the technologies in the charter of this working group or because in some cases (e.g. Credentials Issuance) the solutions are not technological.

7.1 Security Challenges

7.1.1 C-05: Non-Repudiation

Definition: Non-repudiation: A security service that provides protection against false denial of involvement in a communication.

Explanation: Protection against false denial of an action associated with a Web service message. Non-repudiation technologies do not prevent repudiation, but rather provide evidence that may be used by a third party to resolve disputes.

Threat association: Accountability related threats along with threats associated with C-01, C-02 and C-03 must be addressed relative to this challenge and needs to be discussed further.

7.1.2 C-06: Credentials Issuance

Definition: Credential(s): Data that is transferred or presented to establish either a claimed identity or the authorizations of a system entity.

Explanation: The process of initially providing a principal with a means of identifying itself, via online or offline mechanisms. Traditionally, “issuance” refers only to certificates, but here it is used for any information furnished by an authority that is willing to vouch for the principal. We believe that this security challenge is out of scope.

Creation of a credential via transformation from an existing credential to an equivalent one in another format is not issuance in the sense of this section.

Threat association: Out of scope

7.2 Threats

The following threats are considered out of scope for Basic Security Profile. However, these are real threats that need to be considered in any secure application or architecture. There are well-known approaches to addressing these threats that are not documented here.

Note that out of scope threats are designated as T(OOS)-XX.

ID	Name	Description

ID	Name	Description
T(OOS)-01	Key Attack / Weak Algorithm	The algorithm chosen is subject to attacks and/or the key(s) can be compromised. This covers a variety of attacks. Most of these have to do with details of the implementation or operational procedures, which is the reason for considering them to be outside the scope of a specification profile. However some aspects of profiles, e.g. selection of cryptographic algorithms, would be relevant to this threat. Here as elsewhere there are two levels: some parameter settings would be universally considered insecure, e.g. null encryption algorithm. In other cases, the choice would be a matter of local policy. For example, some organizations consider a 1024 bit RSA key adequately strong and others do not. Still others consider it satisfactory for some uses and not others.
T(OOS)-02	Traffic Analysis	By analyzing aspects of the messages such as its source, destination, size, frequency, etc., determinations can be made about potential contents (e.g. it is determined that one company may be trying to buy another). This has many subtle forms. For example, during WW II, Russian scientists deduced that the Americans were building an Atomic Bomb, because the physicists in question had stopped publishing papers.
T(OOS)-03	Host Penetration/Access	Information is obtained by compromising a computer system (e.g. unauthorized access to a computer). Any threat analysis must assume some part of the system is secure. This is called the Trusted Computing Base (TCB). If there is no TCB, it is not possible to conclude anything about the behavior of the system, since presumably an attacker could modify its behavior at will. Thus, in a sense, this threat is out of scope of ANY design or specification, although certainly not out of scope of implementation and operations.
T(OOS)-04	Network Penetration/Access	Information is obtained by compromising a computer network (e.g. unauthorized access to an internal network). This threat presumes a topological approach to security, e.g. firewalls or security gateways. If appropriately strong mechanisms are used on an end-to-end basis, network attacks are reduced to denial-of-service. Thus this threat is out of scope because it is essentially equivalent to the standard assumption of an untrusted network.
T(OOS)-05	Timing	By analyzing the time it takes to perform an action, information can be deduced (e.g. validity of a username, or key information). This is out of scope because it is an implementation issue rather than a specification issue. However, it should be noted that some published cryptographic timing attacks require timing measurements which are much smaller that the average variability of latency in typical networks and thus not of practical concern.

ID	Name	Description
T(OOS)-06	Covert Channels	Information is conveyed outside of a secure perimeter by means of secret communication paths (e.g. by toggling an externally visible flag, secret information is conveyed). This threat is usually only consider seriously in military or intelligence environments. Typically the engineering approach taken is not to eliminate the channel, but to reduce its bandwidth to the point of being useless.
T(OOS)-07	Message Archives	By penetrating the queue of a store-and-forward SOAP intermediary, or the store of an archival system, information about a message can be discovered (e.g. a message in a store and forward queue can be discovered which otherwise wouldn't have been seen). Note that in many circumstances this is a variation on T(OOS)-03. The main reason for calling out this threat separately is because end-to-end message protection measures can counter it, whereas hop-by-hop measures cannot.
T(OOS)-08	Network Spoofing	A message is sent which appears to be from another machine (e.g. BadGuy sends a message which appears as though it is from GoodGuy). Comments similar to those under T(OOS)-04 apply here. If the message does not reach the application, there is little a profile of a specification can have to say about it. If it does reach the application, it is essentially the same as T-03 and T-05.
T(OOS)-08	Trojan Horse	Information is secretly passed along with the message that plants a Trojan horse (e.g. a message is added which is detected by planted software which causes special behaviors to occur).
T(OOS)-09	Virus	Information is secretly passed along with the message that plants a virus (e.g. a message is added which is detected by planted software which causes special behaviors to occur). Note that this is a variation on T-08. Viruses are usually planted by action of unsuspecting user or occasionally program flaw that triggers execution without user action. This can be contrasted with a Worm, which spreads itself autonomously without user action. Worms typically execute other threats found in this table in automated fashion. Some authorities have abandoned the distinction among various programmatic threats and use the term “malware” to cover all types.
T(OOS)-10	Tunneling	Information is secretly passed along with the message (e.g. a message is added which is detected by planted software which causes special behaviors to occur). Note that this is a variation on T-01.

ID	Name	Description
T(OOS)-11	Denial of Service	Silver Bullet: `specific messages or` command sequences causes failure. Almost invariably a result of implementation error, not design error. (Note that this can also result in a system or application compromise instead of merely a Denial of Service.) Addressing this threat is outside of the scope of a profile.
T(OOS)-12	Denial of Service	Flooding: Sheer volume of message traffic overloads some critical resource, typically server or network link bandwidth. This is usually a configuration issue not a design issue. If the bogus traffic is truly indistinguishable from legitimate traffic there may be no defense. It is important to try to • detect that an attack is occurring • determine the true source.
T(OOS)-13	Repudiation	A message is sent and then the sender denies having sent it. Achieving non-repudiation requires both technical and business aspects since a party may always claim a disconnect with the technology ("the software did it, not me, I didn't know").Public Key cryptographic systems have a special property that cannot be achieved by secret key systems without the use of a trusted third party. The property is that it is possible for a party to be able to verify something e.g. a digital signature, without being able to produce it themselves. When this technical property was first observed, it was called ”non-repudiation”. Much later it became widely believed that non-repudiation was a well-established legal concept (It is not.) and very desirable for electronic commerce. The confusion between the technical and legal meanings of this term continues.
T(OOS)-14	Incorrect implementation	If an error is made in implementation of the security protecting a Web service, an attacker could compromise the service by exploiting this security weakness. For example, a signed SOAP message might be susceptible to a certificate substitution attack, which would allow an attacker to modify a message or attach incorrect claims to it. Such threats are out of scope of the profile, as is explicit description of best practices to avoid potential security pitfalls.

ID	Name	Description
T(OOS)-15	Poorly designed Web services	Simply securing Web services does not secure an application as a whole. A poorly designed service, such as one that is susceptible to SQL injection attacks, or spawns a shell that accepts parameters from a SOAP message, can be compromised even though the transaction itself is considered secure. Such threats are naturally out of scope of this profile.

Table 4: Out of Scope Threats

8 Acronyms

HTTP – Hypertext Transfer Protocol

HTTPS – Hypertext Transfer Protocol Secure

IETF – Internet Engineering Task Force

MD5 – one Message-Digest algorithm (RFC-1321)

MEP – Message Exchange Pattern

MIME – Multipurpose Internet Mail Extensions

OASIS – not an acronym

OOS – Out Of Scope

REL – Rights Expression Language

RFC – Request for Comment (Used by IETF)

SAML – Security Assertions Markup Language

SCM – Supply Chain Management; the WS-I Sample Application for 1.0

SHA – Secure Hash Algorithm

SOAP - Simple Object Access Protocol

SSL – Secure Sockets Layer

TLS – Transport Layer Security

WS-Security – OASIS SOAP Message Security specifications

XML – Extensible Markup Language

X.509 – An ITU (International Telecommunication Union) standard for “certificates” Also known as ISO/IEC 9594-8:1988

9 References

1. [BP 1.0] Basic Profile 1.0.
   http://www.ws-i.org/Profiles/BasicProfile-1.0.html
2. [SOAP 1.1] Simple Object Access Protocol (SOAP) 1.1
   http://www.w3.org/TR/2000/NOTE-SOAP-20000508
3. [SOAP 1.2] SOAP Version 1.2 Part 1: Messaging Framework
   http://www.w3.org/TR/soap12-part1
4. [RFC 2616] Hypertext Transport Protocol – HTTP 1.1
   http://www.ietf.org/rfc/rfc2616.txt
5. [RFC 2617] HTTP Authentication: Basic and Digest Access Authentication, June 1999, Obsoletes RFC 2069
   http://www.ietf.org/rfc/rfc2617.txt
6. [RFC 2246] The TLS Protocol. Version 1.0
   http://www.ietf.org/rfc/rfc2246.txt
7. [RFC 2828] Internet Security Glossary
   http://www.ietf.org/rfc/rfc2828.txt
8. [BPSA UsageScenarios] WS-I Usage Scenarios
   http://www.ws-i.org/SampleApplications/SupplyChainManagement/2003-12/UsageScenarios-1.01.pdf
9. [SwA] Soap With Attachments
   http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211
10. [AP 1.0] AttachmentsProfile 1.0
    http://www.ws-i.org/Profiles/Basic/2003-08/AttachmentsProfile-1.0.pdf
11. [WSS 1.0] Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)

    http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf
12. [UTP 1.0] Web Services Security Username Token Profile 1.0

    http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf
13. [X509 1.0] Web Services Security X.509 Certificate Token Profile

    http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf
14. [SAML 1.0] Web Services Security: SAML Token Profile

    http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0.pdf
15. [REL 1.0] Web Services Security Rights Expression Language (REL) Token Profile

    http://docs.oasis-open.org/wss/oasis-wss-rel-token-profile-1.0.pdf

10 Informative References

1. [OWASP] The Open Web Application Security Project (http://easynews.dl.sourceforge.net/sourceforge/owasp/OWASPWebApplicationSecurityTopTen-Version1.pdf)
2. [SCM-UC] Supply Chain Management Use Cases (http://ws-i.org/SampleApplications/SupplyChainManagement/2002-11/SCMUseCases-0.18-WGD.pdf)
3. [SCM-US] Supply Chain Management Usage Scenarios (http://ws-i.org/SampleApplications/SupplyChainManagement/2002-11/UsageScenarios-1.00-CRD-02a.pdf)
4. [WSA] W3C Web Services Architecture Usage Scenarios (http://www.w3.org/TR/2002/WD-ws-arch-scenarios-20020730/)
5. Stallings, William. *Cryptography and Network Security: Principles and Practice (3rd Edition)*, Prentice Hall 2002
6. Fisch, Eric A and White, Gregory B. *Secure Computers and Networks: Analysis, Design, and Implementation*, CRC Press, 1999
7. Kaufman, Charlie and Perman, Radia and Speciner, Mike. *Network Security: Private Communication in a Public World*, Prentice Hall, 2002
8. Ford, Warwick and Baum, Michael S. *Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption (2nd Edition)*, Prentice Hall, 2000
9. *Schneier, Bruce. Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition*. John Wiley & Sons. 1995