Web Services Security: Receipt Token Profile

Proposal Draft 06, 5 May 2003

Document identifier:
web-services-receipt-token-profile-06 (PDF, Word)

Location:
http://schemas.reactivity.com/2003/05/web-services-receipt-token-profile-06.pdf

Editor:
Eric L. Gravengaard, Reactivity, Inc. <eric@reactivity.com>

Contributors:
Grant Goodale, Reactivity
Michael Hanson, Reactivity
Brian Roddy, Reactivity
Dan Walkowski, Reactivity

Abstract:
This specification defines a method for requesting and sending message disposition receipts inside SOAP message headers. This specification makes use of the Web Services Security: SOAP Message Security and XML Digital Signature specifications.

Status:
This document is a proposal draft. Comments should be directed to the editor.

Table of Contents

1 Introduction 3
2 Terminology 3
2.1 Namespaces 3
2.2 Glossary of Terms 4
3 Non-Normative Requirements 4
4 ReceiptRequest Element 4
4.1 ReceiptTo Element 6
5 Receipt Element 7
6 Use of Digital Signatures 8
6.1 SignatureRequest Element 8
6.2 SignatureResponse Element 9
6.3 Non-Normative Processing Model 10
7 Global Attributes 10
7.1 ReceiptFormat Attribute 10
7.2 CorrelationId Attribute 10
8 Error Handling 11
9 Security Considerations 11
10 Non-Normative Example 12
10.1 Simple Example 12
10.1.1 Request 12
10.1.2 Response 12
10.2 Signed Example 12
10.2.1 Request 12
10.2.2 Response 13
10.3 Full Example with Security Precautions 13
10.3.1 Request 13
10.3.2 Response 15
11 References 16
11.1 Normative 16
11.2 Non-Normative 17
Appendix A. Revision History 18
Appendix B. Notices 19

1 Introduction

The Web Services Security: SOAP Message Security specification **[WSS]** defines the usage of XML Digital Signatures within a SOAP header element to prove the integrity of a SOAP message. While this is useful in the context of non-repudiation to the receiver, it does nothing to guarantee to the sender that the message was delivered properly and without modification. Similarly, when the SOAP requestor receives the SOAP response message there is no way of proving that the SOAP response was generated after receiving and processing the SOAP request.

This specification extends the use of XML Digital Signature in the context of WSS: SOAP Message Security to allow senders of SOAP messages to request message disposition notifications that may optionally be signed to prove that the receiver received the SOAP message without modification. The specification also defines a method for embedding SOAP message dispositions in a SOAP message header. This specification constitutes a protocol for voluntary non-repudiation of receipt that when used systematically provides cryptographic proof of both parties participation in a transaction. This specification does not define any mechanism to prove receipt of a message by a non-conformant implementation.

2 Terminology

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, NOT RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in **[RFC2119]**.

This specification is designed to work with the general SOAP message structure and message processing model, and should be applicable to any version of SOAP. The current SOAP 1.2 namespace URI is used herein to provide detailed examples, but there is not intention to limit the applicability of this specification to a single version of SOAP.

2.1 Namespaces

The following XML namespace URI MUST be used by implementations of this specification is as follows:

```
http://schemas.reactivity.com/2003/04/wsnr
```

The following namespaces are used in this document:

Prefix	Namespace
S12	http://www.w3.org/2002/12/soap-envelope
ds	http://www.w3.org/2000/09/xmldsig#
wss	http://schemas.xmlsoap.org/ws/2003/03/secext
wsu	http://schemas.xmlsoap.org/ws/2002/07/utility
xs	http://www.w3.org/2001/XMLSchema

2.2 Glossary of Terms

Actor

An *actor* is any processor, the requestor, ultimate destination, or SOAP intermediary, which receives and processes a SOAP message.

Integrity

Integrity is the property that data has not been modified.

Message Disposition Notification

Message Disposition Notification is a message reporting the status of a message. It conveys information about whether the message was received and possibly if its integrity was preserved in transit.

Signature

A *signature* is a value computed with a cryptographic algorithm and bound to data in such a way that intended recipients of the data can use the signature to verify that the data has not been altered since it was signed by the signer **[XMLDSIG]**.

SOAP Intermediary

A *SOAP intermediary* is an application that is capable of both receiving and forwarding SOAP messages.

SOAP Message Requestor

The *SOAP Message Requestor* is the originator of a SOAP Message and the client in the HTTP Protocol binding defined in SOAP 1.1**[SOAP11]**.

SOAP Message Responder

The *SOAP Message Responder* is the ultimate receiver of a SOAP Message and the server in the HTTP Protocol binding defined in SOAP 1.1**[SOAP11]**.

3 Non-Normative Requirements

This specification was designed to satisfy four requirements:

1. SOAP Message Requestors must be able to request a receipt for the SOAP Message that is being transmitted.
2. SOAP Message Responders must be able to send a receipt for a SOAP Message that requests one, either embedded in the SOAP Response or in another message.
3. SOAP Message Requestors must be able to specify what elements of the SOAP Message they wish to have signed by the SOAP Message Responder.
4. SOAP Message receipts must be able to convey a signature for the elements that were requested to be signed by the SOAP Message Requestor

4 ReceiptRequest Element

An actor uses the **<ReceiptRequest>** element to request that a subsequent actor send one or more receipts for the message in which the **<ReceiptRequest>** is placed. The **<ReceiptRequest>** element MUST be placed within a **<wss:Security>** element that has the **role** attribute set such that the appropriate actor will process the **<ReceiptRequest>**. Each **<ReceiptRequest>** element specifies the type of receipt and a set of one or more destinations

for the receipt to be sent. Multiple **<ReceiptRequest>** elements MAY be placed within the same **<wss:Security>** element if multiple types of receipts are requested.

The syntax for this element is as follows:

```
<xs:element name="ReceiptRequest">
   <xs:complexType>
       <xs:sequence>
           <xs:element name="ReceiptTo" maxOccurs="unbounded">
               …
           </xs:element>
           <xs:element ref="SignatureRequest" minOccurs="0"/>
           <xs:element ref="wsu:Timestamp" minOccurs="0"/>
       </xs:sequence>
       <xs:attribute ref="ReceiptFormat" use="required"/>
       <xs:attribute ref="CorrelationId" use="optional"/>
       <xs:attribute ref="wsu:Id" use="optional"/>
       <xs:attribute ref="S12:mustUnderstand" use="optional"/>
       <xs:anyAttribute/>
   </xs:complexType>
</xs:element>
```

/ReceiptRequest/@ReceiptFormat

The **ReceiptFormat** attribute designates the type of receipt that is requested. The attribute is of type **xs:anyURI**. The attribute is required. The legal values of **ReceiptFormat** are described in section 7.1. When using either the **generalReceipt** or the **signedReceipt** formats, the **<ReceiptRequest>** MAY contain a **<wsu:TimeStamp>** element. When using the **signedReceipt** format, the **<ReceiptRequest>** MUST contain a **<SignatureRequest>** element. Conformant implementations MUST be able to process both formats.

/ReceiptRequest/@CorrelationId

The optional **CorrelationId** attribute is used in the **<ReceiptRequest>** to specify a unique identifier for the **<ReceiptRequest>**. When a **CorrelationId** is included in the request, actors responding with receipts MUST include the **CorrelationId** of the request to allow the requestor to match the request to the receipt. The type and recommended usage of **CorrelationId** is specified in section 7.2.

/ReceiptRequest/@wsu:Id

The optional **wsu:Id** attribute is used to specify a unique identifier for the **<ReceiptRequest>** element that can be used to reference the element. It is RECOMMENDED that this attribute be used to allow the **<ReceiptRequest>** to be signed using an XML Digital Signature.

/ReceiptRequest/@S12:mustUnderstand

The optional **S12:mustUnderstand** attribute allows requestors to specify that a SOAP Fault MUST be returned if the actor cannot process the **<ReceiptRequest>** element. The **mustUnderstand** attribute MUST be in the same namespace of the root **<Envelope>** element.

/ReceiptRequest/@any

The **any** attribute is included in the schema for the purpose of satisfying the multiple namespaces in which the **mustUnderstand** attribute may be described. This is the ONLY permitted use of this schema extension.

/ReceiptRequest/ReceiptTo

One or more **<ReceiptTo>** elements are required to indicate where the receipt should be sent. When multiple **<ReceiptTo>** elements are present, a copy of the **<Receipt>** MUST be sent to each of the targets specified unless the **Required** element is set to false. The syntax for the **<ReceiptTo>** element is described in section 4.1.

/ReceiptRequest/SignatureRequest

The **<SignatureRequest>** element MUST be present if and only if the **ReceiptFormat** attribute equals **signedReceipt**. The syntax of this element is described in section 6.1.

/ReceiptRequest/wsu:TimeStamp

The optional **<wsu:TimeStamp>** element MAY be included to indicate the creation time of the **<ReceiptRequest>** element. If the element is included, conformant implementations MUST return a SOAP Fault if the creation time is in the future or if the expiration time is in the past.

4.1 ReceiptTo Element

The **<ReceiptTo>** element is used to convey information on how a responding actor should send a receipt to the receipt requestor. The syntax for this element is as follows:

```
<xs:element name="ReceiptTo" maxOccurs="unbounded">
   <xs:complexType>
          <xs:attribute name="Target" type="xs:anyURI" use="optional"
default="http://schemas.reactivity.com/wsnr/2003/04/response"/>
          <xs:attribute name="Required" type="xs:boolean" use="optional"
default="1"/>
          <xs:attribute name="ReceiptAddress" type="xs:anyURI" use="optional"/>
          <xs:attribute ref="S12:role" use="optional"
default="http://www.w3.org/2002/12/soap-envelope/role/ultimateReceiver"/>
          <xs:anyAttribute/>
   </xs:complexType>
</xs:element>
```

/ReceiptRequest/ReceiptTo/@Target

The optional **Target** attribute specifies where the **<Receipt>** should be sent. If this attribute is omitted, then the default value, **response,** should be assumed. There can be only one **<ReceiptTo>** element that omits the **Target** attribute or explicitly has **Target** equal to **response**. The **response** option is only available for the request SOAP message in a two-message request-response context. Clearly, a **<ReceiptTo>** element within an HTTP response MUST NOT have **Target** equal to **response** and MUST NOT omit the **Target** attribute. Legal values for **Target** are shown below:

Short name	Long name / Description
response (default)	http://schemas.reactivity.com/2003/04/wsnr/response
	The **<Receipt>** should be included in the **<wss:Security>** header of the response SOAP message.
HTTPSOAP	http://schemas.reactivity.com/2003/04/wsnr/HTTPSOAP
	A SOAP message with a **<Receipt>** in the **<wss:Security>** header should be sent to the URL indicated in the **ReceiptAddress** attribute.
SMTPSOAP	http://schemas.reactivity.com/2003/04/wsnr/SMTPSOAP
	A SOAP message with a **<Receipt>** in the **<wss:Security>** header should be sent to the email address indicated in the **ReceiptAddress** attribute.

If the value of Target is not response, then a **ReceiptAddress** attribute MUST be included in the **<ReceiptTo>** element.

/ReceiptRequest/ReceiptTo/@Required

The optional **Required** attribute is used to indicate if the **<Receipt>** is required by the requestor or if is optional. If the Required attribute is set to true, then the responding actor MUST send either a **<Receipt>** or a SOAP Fault. If the **Required** attribute is set to false, then the

responding actor MAY return a **<Receipt>** depending on its own security policies. The default value is true if the attribute is omitted.

/ReceiptRequest/ReceiptTo/@ReceiptAddress

The **ReceiptAddress** attribute MUST be specified if the **Target** attribute equals either **HTTPSOAP** or **SMTPSOAP**. If the value of **Target** is **HTTPSOAP**, then the value of the **ReceiptAddress** attribute MUST be a HTTP/S URL where a SOAP message containing a **<wss:Security>** element containing a **<Receipt>** MUST be sent using the HTTP POST protocol. If the value of **Target** is **SMTPSOAP**, then the value of the **ReceiptAddress** attribute MUST be a **mailto:** URL that specifies an SMTP address where a SOAP message containing a **<wss:Security>** element containing a **<Receipt>** MUST be sent.

/ReceiptRequest/ReceiptTo/@S12:role

The optional **S12:role** attribute specifies the value of role in the **<wss:Security>** header that MUST contain the **<Receipt>**. Compliant implementations MUST either use an existing **<wss:Security>** header with the corresponding **S12:role** attribute or insert a new **<wss:Security>** header with the appropriate **S12:role** attribute. The **S12:role** attribute SHOULD be specified in the same namespace as the corresponding outer **<Envelope>** to enable the receiver to properly interpret this value and respond appropriately. The default value if the attribute is not specified is **S12:ultimateReceiver**.

/ReceiptRequest/ReceiptTo/@any

The **any** attribute is included in the schema for the purpose of satisfying future namespaces in which the **role** attribute may be described. This is the ONLY permitted use of this schema extension.

5 Receipt Element

The **<Receipt>** element is used to respond to a **<ReceiptRequest>**. It conveys the fact that the message to which the **<ReceiptRequest>** was attached to was received but not that it was interpreted correctly or processed. Therefore, the **<Receipt>** may be issued even when there was a problem processing the message that contained the **<ReceiptRequest>** so long as the **<ReceiptRequest>** was processed correctly. For example, a **<Receipt>** element could be returned in the event of a SOAP Fault. **<Receipt>** elements MUST be placed within a **<wss:Security>** header element of a SOAP message. The syntax for the **<Receipt>** element is as follows:

```
<xs:element name="Receipt">
   <xs:complexType>
          <xs:sequence>
                 <xs:element ref="SignatureResponse" minOccurs="0"/>
                 <xs:element ref="wsu:Timestamp" minOccurs="0"/>
          </xs:sequence>
          <xs:attribute ref="ReceiptFormat" use="required"/>
          <xs:attribute ref="CorrelationId" use="optional"/>
          <xs:attribute ref="wsu:Id" use="optional"/>
   </xs:complexType>
</xs:element>
```

/Receipt/@ReceiptFormat

The **ReceiptFormat** attribute designates what type of **<Receipt>** is being sent. The attribute is of type **xs:anyURI**. The attribute is required. The legal values of **ReceiptFormat** are described in section 7.1. When using either the **generalReceipt** or the **signedReceipt** formats, the **<Receipt>** MAY contain a **<wsu:TimeStamp>** element. When using the **signedReceipt** format, the **<Receipt>** MUST contain a **<SignatureResponse>** element.

/Receipt/@CorrelationId

The optional **CorrelationId** attribute is used in the **<Receipt>** to specify the unique identifier used to identify the **<ReceiptRequest>**. When a **CorrelationId** is included in the request, actors responding with receipts MUST include the **CorrelationId** of the request to allow the requestor to match the request to the receipt. The type and recommended usage of **CorrelationId** is specified in section 7.2.

/Receipt/@wsu:Id

The optional **wsu:Id** attribute is used to specify a unique identifier for the **<Receipt>** element that can be used to reference the element. It is RECOMMENDED that this attribute be used to allow the **<Receipt>** to be signed using an XML Digital Signature.

/Receipt/SignatureResponse

The **<SignatureResponse>** element MUST be present if and only if the **ReceiptFormat** attribute equals **signedReceipt**. The syntax of this element is described in section 6.2.

/Receipt/wsu:TimeStamp

The optional **<wsu:TimeStamp>** element MAY be included to indicate the creation time of the **<Receipt>** element. The use of **<wsu:Expires>** element has no meaning in this context.

6 Use of Digital Signatures

Using digital signatures adds further capabilities to the use of **<Receipts>**. XML digital signatures allow the receiving party to verify the integrity of the signed data and the identity of the signing party. The format and processing instructions for XML digital signatures have been defined by the W3C **[XMLDSIG]**. However a single signature generated by a single actor cannot

signature has been divided into two halves: that which should be signed, and the actual signature value. The XML Digital Signature specification defines several components that can be used to describe both of these components in greater detail. This specification defines two new elements, **<SignatureRequest>** and **<SignatureResponse>**, to hold the sub-elements of a **<ds:Signature>** that represent a request for signature and a response.

When combined, the sub-elements of a **<SignatureRequest>** and **<SignatureResponse>**, form all the subcomponents of a **<ds:Signature>**.

6.1 SignatureRequest Element

The **<SignatureRequest>** element is used to request a digital signature that covers a dataset specified by the **<ds:SignedInfo>** sub-element. The syntax for this element is as follows:

```
<xs:element name="SignatureRequest">
   <xs:complexType>
      <xs:sequence>
         <xs:element ref="ds:SignedInfo"/>
         <xs:element ref="ds:Object" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute ref="wsu:Id" use="optional"/>
      <xs:anyAttribute/>
   </xs:complexType>
</xs:element>
```

/SignatureRequest/@wsu:Id

The optional **wsu:Id** attribute is used to specify a unique identifier for the **<SignatureRequest>** element that can be used to reference the element. It is intended to

allow the element to be signed using an XML Digital Signature if the element is not used in the context of a **<ReceiptRequest>**.

/SignatureRequest/@any

The **any** attribute is used to allow this element to be used in contexts other than **<ReceiptRequest>**. When used in the context of a **<ReceiptRequest>** element, no other attributes are defined for the **<SignatureRequest>** element.

/SignatureRequest/ds:SignedInfo

The **<ds:SignedInfo>** element is used to convey information about what dataset the requestor would like signed. The **<ds:SignedInfo>** element will also convey information about the state of the dataset at the time of request because it will contain one or more **<ds:Reference>** elements with a corresponding **<ds:DigestValue>**. The recipient of the **<SignatureRequest>** SHOULD verify that the **<ds:SignedInfo>** is still valid and then compute a **<ds:SignatureValue>**. The element is a required sub-element of **<SignatureRequest>**.

/SignatureRequest/ds:Object

Zero or more **<ds:Object>** elements can be used to hold additional data for the signature to be computed over. It is allowed here only to allow for the reuse of the **<SignatureRequest>** element in other contexts. Since no processing will occur on **<ds:Object>** elements, it is NOT RECOMMENDED that they be used in the context of a **<ReceiptRequest>**.

6.2 SignatureResponse Element

The **<SignatureResponse>** element is used to respond to a **<SignatureRequest>**. It contains both the **<ds:SignatureValue>** and optional **<ds:KeyInfo>** elements that together with the sub-elements of the original **<SignatureRequest>** comprise a complete **<ds:Signature>**. The syntax for the **<SignatureResponse>** element is as follows:

```
<xs:element name="SignatureResponse">
   <xs:complexType>
          <xs:sequence>
                 <xs:element ref="ds:SignatureValue"/>
                 <xs:element ref="ds:KeyInfo" minOccurs="0"/>
          </xs:sequence>
          <xs:attribute ref="wsu:Id" use="optional"/>
          <xs:anyAttribute/>
   </xs:complexType>
</xs:element>
```

/SignatureResponse/@wsu:Id

The optional **wsu:Id** attribute is used to specify a unique identifier for the **<SignatureResponse>** element that can be used to reference the element. It is intended to allow the element to be signed using an XML Digital Signature if the element is not used in the context of a **<Receipt>**.

/SignatureResponse/@any

The **any** attribute is used to allow this element to be used in contexts other than **<Receipt>**. When used in the context of a **<Receipt>**, no other attributes are defined for the **<SignatureResponse>** element.

/SignatureResponse/ds:SignatureValue

The required **<ds:SignatureValue>** element conveys the value of the cryptographic signature that covers the **<ds:SignedInfo>** element from the **<SignatureRequest>**.

/SignatureResponse/ds:KeyInfo

The optional **<ds:KeyInfo>** element conveys information about the key used to compute the **<ds:SignatureValue>**. The element can contain any legal values as specified in the XML Digital Signature and Web Services Security specifications. It is RECOMMENDED that the **<ds:KeyInfo>** element contain a **<wss:SecurityTokenReference>** and that the key be prepended to the enveloping **<wss:Security>** header before the **<Receipt>** element.

6.3 Non-Normative Processing Model

To generate a **<SignatureRequest>**, the SOAP Message Requestor should first generate the SOAP envelope including the data contained in the **<S12:Body>**. Then the Requestor should create a **<ds:SignedInfo>**, including all references, for the data that the Responder should validate and for which it should respond with a disposition notification. Then the requestor should prepend the **<ReceiptRequest>** with the **<SignatureRequest>** into the applicable **<wss:Security>** header. A signature may then be created that references the **<ReceiptRequest>** and prepended to the **<wss:Security>** header.

When processing a **<ReceiptRequest>**, the Responder should first validate that the **<ds:SignedInfo>** is valid and that all of the **<ds:DigestValue>** elements are still valid. If they are not, then the Responder cannot attest to having received the data for which the Requestor asked to receive a disposition notification and therefore a proper receipt cannot be generated. If the digests are valid, then the Responder should calculate a **<ds:SignatureValue>**. This can then be inserted into a **<Receipt>** in the appropriate **<wss:Security>** header of the SOAP response message. A signature may then be created that references the <Receipt> and is prepended to the **<wss:Security>** header.

7 Global Attributes

The following attributes are used by multiple elements defined in this specification:

7.1 ReceiptFormat Attribute

The **ReceiptFormat** attribute designates the format of the **<Receipt>** or **<ReceiptRequest>** element. The attribute is of type **xs:anyURI**. The attribute is required for both **<Receipt>** and **<ReceiptRequest>** elements. There are two legal values for this attribute:

Short name	Long name	Description
generalReceipt	http://schemas.reactivity.com/2003/04/wsnr/generalReceipt	This format is for general unsigned receipts.
signedReceipt	http://schemas.reactivity.com/2003/04/wsnr/signedReceipt	This format is for signed receipts.

7.2 CorrelationId Attribute

The **CorrelationId** attribute is an **xs:string** that can be used to uniquely identify a pair of **<ReceiptRequest>** and **<Receipt>** elements. It MUST be unique to both the sender and the recipient so that each may log it and reference it later by this value. It is RECOMMENDED that the **CorrelationId** value be formatted as an **urn:uuid [UUID]**. For example:

```
<ReceiptRequest
          CorrelationId="urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
          ReceiptFormat="generalReceipt">
   <ReceiptTo/>
</ReceiptRequest>
```

8 Error Handling

If the Responder does not understand how to process a **<ReceiptRequest>** then the Responder MUST return a SOAP Fault or stop processing. The **faultcode** for this class of error is **S12:MustUnderstand**.

When using the **signedRequest** format, if the Responder cannot verify that the **<ds:SignedInfo>** references are valid, then the Responder MUST NOT send a receipt. The Responder MUST return a SOAP Fault or stop processing. The **faultcode** for this class of error is **wsnr:InvalidSignedInfo**.

9 Security Considerations

There are three main security considerations when using this specification for secure non-repudiation. First, both the receipt requestor and the receipt generator should keep secure records of all message traffic. This is important because the complete signature is distributed across both the request and the receipt and can only be verified when both pieces are present. Only when both sides of an exchange log both pieces can both parties make any guarantee of message disposition. Schneier and Kelsey present a cryptographic method for secure logging in their 1999 paper **[Schneier]**.

Second, both the **<ReceiptRequest>** and the **<Receipt>** elements should be signed. This allows the receiving party to know that neither the **<ReceiptRequest>** nor the **<Receipt>** were tampered with en route. In the case of the **<ReceiptRequest>**, this guarantees that the **<ds:SignedInfo>** element was not changed to remove a key element from the dataset used for the computation of the signature value.

Third, the trust relationship between two parties impacts the level of acceptance each party should have for the other party's notion of time. As previously recommended, the **<ReceiptRequest>** and **<Receipt>** elements should include a **<wsu:Timestamp>** element indicating the time the encapsulating element was generated. If the encapsulating element is digitally signed following the method described in Section 6, this timestamp may be taken at face value in communications between parties with a medium to high degree of trust.

In communications between parties with a low degree of trust, a trusted digital time stamping service capable of producing digitally signed timestamps in a format understood by both parties should be used. The signed timestamp should at a minimum contain the digest of the **<ReceiptRequest>** element and all elements referenced within the receipt request. In any event, timestamps containing future times or times that differ from the receiving party's notion of the current time should be treated as highly suspect.

10 Non-Normative Example

10.1 Simple Example

10.1.1 Request

```
<wsse:Security>
   <ReceiptRequest ReceiptFormat="generalReceipt" CorrelationId="33485">
          <ReceiptTo Required="true" Target="response"/>
          <wsu:Timestamp>
                <wsu:Created>2003-03-11T16:30:17Z</wsu:Created>
          </wsu:Timestamp>
   </ReceiptRequest>
</wsse:Security>
```

10.1.2 Response

```
<wsse:Security>
   <Receipt ReceiptFormat="generalReceipt" CorrelationId="33485">
          <wsu:Timestamp>
                <wsu:Received>2003-03-11T16:33:43Z</wsu:Received>
          </wsu:Timestamp>
   </Receipt>
</wsse:Security>
```

10.2 Signed Example

10.2.1 Request

```
<S:Envelope xmlns:S="...">
   <S:Header>
       <wsse:Security>
           <wsnr:ReceiptRequest ReceiptFormat="signedReceipt"
               Role="ultimateReceiver" CorrelationID="theID"
               S:mustUnderstand="1">
               <wsnr:ReceiptTo Target="response">
                   <wsnr:SignatureRequest>
                       <ds:SignedInfo>
                           <ds:CanonicalizationMethod Algorithm="#c14n"/>
                           <ds:SignatureMethod Algorithm="#hmac-sha1"/>
                           <ds:Reference URI="#body">
                               <ds:DigestMethod Algorithm="#sha1"/>
                           </ds:Reference>
                           <ds:Reference URI="#timestamp">
                               <ds:DigestMethod Algorithm="#sha1"/>
                           </ds:Reference>
                       </ds:SignedInfo>
                   </wsnr:SignatureRequest>
               </wsnr:ReceiptTo>
               <wsu:Timestamp wsu:Id="timestamp">
                   <wsu:Created>2003-03-11T08:42:00Z</wsu:Created>
               </wsu:Timestamp>
           </wsnr:ReceiptRequest>
       </wsse:Security>
   </S:Header>
   <S:Body>
       <MyRequest wsu:Id="body"/>
   </S:Body>
</S:Envelope>
```

10.2.2 Response

```
<S:Envelope xmlns:S="...">
   <S:Header>
       <wsse:Security S:Role="ultimateReceiver">
           <wsse:BinarySecurityToken wsu:Id="#theCert"
               EncodingType="Base64Binary">
                MIIEZzCCA9CgAWIQEmtJZco...
       </wsse:BinarySecurityToken>
           <wsnr:Receipt ReceiptFormat="signedReceipt"
               CorrelationID="theID">
               <wsnr:SignatureResponse>
                   <ds:SignatureValue>
                       ABCDEFG1234567890...
                   </ds:SignatureValue>
                   <ds:KeyInfo>
                       <wsse:SecurityTokenReference>
                           <wsse:Reference URI="#theCert"/>
                       </wsse:SecurityTokenReference>
                   </ds:KeyInfo>
               </wsnr:SignatureResponse>
               <wsu:Timestamp>
                   <wsu:Received>2003-03-11T08:42:12Z</wsu:Received>
               </wsu:Timestamp>
           </wsnr:Receipt>
       </wsse:Security>
   </S:Header>
   <S:Body>
       <MyResponse/>
   </S:Body>
</S:Envelope>
```

10.3 Full Example with Security Precautions

The following example shows the non-normative recommended usage of this specification to securely request and send a receipt using the **`signedReceipt`** format.

10.3.1 Request

The SOAP Message Requestor generates a **`<ds:SignedInfo>`** element that references and digests the **`<S12:Body>`** and the **`<wsu:Timestamp>`** elements. The **`<ReceiptRequest>`** element is then signed. The Requestor does not need to additionally sign the **`<S12:Body>`** element because that is covered by the signed **`<ds:SignedInfo>`** of the **`<SignatureRequest>`**. Any changes to the **`<S12:Body>`** will result in the **`<SignatureRequest>`** becoming invalid and therefore the SOAP Message Responder will detect the loss of integrity.

```
<?xml version="1.0" encoding="UTF-8"?>
<S12:Envelope xmlns:wsnr="http://schemas.reactivity.com/2003/04/wsnr/"
xmlns:S12="http://www.w3.org/2002/12/soap-envelope"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/xx/utility"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/xx/secext">
 <S12:Header>
   <wsse:Security>
     <wsse:BinarySecurityToken EncodingType="wsse:Base64Binary"
wsu:Id="RequestorCert">
MIIEfzCCA+igAwIBAgIBAzANBgkqhkiG9w0BAQQFADCB3DELMAkGA1UEBhMCVVMxEzARBgNVBAgT
CkNhbGlmb3JuaWExEDAOBgNVBAcTB0JlbG1vbnQxIDAeBgNVBAoTF1JlYWN0aXZpdHkgVGVzdCBD
b21wYW55MS4wLAYDVQQLEyVSZWFjdGl2aXR5IFRlc3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5MS4w
LAYDVQQDEyVSZWFjdGl2aXR5IFRlc3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5MSQwIgYJKoZIhvcN
AQkBFhVzb21lb25lQHNvbWV3aGVyZS5jb20wHhcNMDIwODI5MDAyMzU5WhcNMDMwODI5MDAyMzU5
WjCBnTELMAkGA1UEBhMCVVMxFjAUBgNVBAgTDU1hc3NhY2h1c2V0dHMxDzANBgNVBAcTBkJvc3Rv
bjESMBAGA1UEChMJQ29tcGFueSBBMRcwFQYDVQQLEw5Pcmdhbml6YXRpb24gQTESMBAGA1UEAxMJ
Q29tcGFueSBBMSQwIgYJKoZIhvcNAQkBFhVzb21lb25lQHNvbWV3aGVyZS5jb20wgZ8wDQYJKoZI
hvcNAQEBBQADgY0AMIGJAoGBALM+RhnZwfT6s1vdFpn+amZ7CvJIfdmXJgCRBzwvczNqdJhFtw14
```

```
NX7Po2YM7vn/nlHw0E3yP3cwKqfHfAzvlhs5TuEXnfvjQAgTvJZYudROc+D1w2QBjCtg/ox/0WNC
wU9eiHuHC3fm5ewCsx/H0WwuIThpOyUbWSl1NFKcJoXBAgMBAAGjggGMMIIBiDAJBgNVHRMEAjAA
MCwGCWCGSAGG+EIBDQQfFh1PcGVuU1NMIEdlbmVyYXRlZCBDZXJ0aWZpY2F0ZTAdBgNVHQ4EFgQU
PvkJwoTrduf/QbKxmPPZRGplls8wggEKBgNVHSMEggEBMIH+gBRsm+JodlO91efBrp8LkN/UC76N
AqGB4qSB3zCB3DELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExEDAOBgNVBAcTB0Jl
bG1vbnQxIDAeBgNVBAoTF1JlYWN0aXZpdHkgVGVzdCBDb21wYW55MS4wLAYDVQQLEyVSZWFjdGl2
aXR5IFRlc3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5MS4wLAYDVQQDEyVSZWFjdGl2aXR5IFRlc3Qg
Q2VydGlmaWNhdGUgQXV0aG9yaXR5MSQwIgYJKoZIhvcNAQkBFhVzb21lb25lQHNvbWV3aGVyZS5j
b22CAQAwCwYDVR0PBAQDAgWgMBMGA1UdJQQMMAoGCCsGAQUFBwMCMA0GCSqGSIb3DQEBBAUAA4GB
AHpycTmqU2cMnlk8lAEgG+WuD6zP5GWqBgdw199J3JuDpfg/1fiF1QhCQJi/53DYO/edogVt276n
2pPcWqoemRnhVjmsGe0GzkQHFP445/++g1RuvOkhXthh2GoGI8P3tzA1wo8F7syJRxSEnTF2jO8E
ZbzPUKlB+TuC3MsRk0gi</wsse:BinarySecurityToken>
     <ds:Signature>
       <ds:SignedInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
         <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-
c14n-20010315"/>
         <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"/>
         <ds:Reference URI="#receiptRequest2328348">
           <ds:Transforms>
             <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
           </ds:Transforms>
           <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
           <ds:DigestValue>wlTM8NLGwt6ZAuM/yX1Cu/1gv3I=</ds:DigestValue>
         </ds:Reference>
       </ds:SignedInfo>

 <ds:SignatureValue>ekkILVSMIageKDVKw5hNpD8F6QBfNlRY5bwzS4lQ/VLiIQxPlULCGHHFIM68
uQMKW2E7wQ9ohiQe
x3aykuRT5HNtpA9BI3lEP2BPSlqOfj161iMzKhKHQxYXlixsg8CdglCjaAy1PDxCQoskF1cgjHOr
U6E7d9Ag9s33HqKGX2Q=</ds:SignatureValue>
       <ds:KeyInfo>
         <wsse:SecurityTokenReference>
           <wsse:Reference URI="#RequestorCert"/>
         </wsse:SecurityTokenReference>
       </ds:KeyInfo>
     </ds:Signature>
     <wsnr:ReceiptRequest ReceiptFormat="signedReceipt" S12:mustUnderstand="true"
wsnr:CorrelationId="urn:uuid:f81d4fde-7dec-11d0-a765-00a0c91e6bf6"
wsu:Id="receiptRequest2328348">
       <wsnr:ReceiptTo/>
       <wsnr:SignatureRequest>
         <ds:SignedInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
           <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-
xml-c14n-20010315"/>
           <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"/>
           <ds:Reference URI="#body2328348">
             <ds:Transforms>
               <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
             </ds:Transforms>
             <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
             <ds:DigestValue>9bKI84lheW6NCbnjltD4ZJi0wZ0=</ds:DigestValue>
           </ds:Reference>
           <ds:Reference URI="#timestamp2328348">
             <ds:Transforms>
               <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
             </ds:Transforms>
             <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
             <ds:DigestValue>uNFkh+T9hVqvjKhmt6lGc90jMkI=</ds:DigestValue>
           </ds:Reference>
         </ds:SignedInfo>
       </wsnr:SignatureRequest>
       <wsu:Timestamp wsu:Id="timestamp2328348">
         <wsu:Created>2003-03-12</wsu:Created>
       </wsu:Timestamp>
```

```
    </wsnr:ReceiptRequest>
  </wsse:Security>
 </S12:Header>
 <S12:Body wsu:Id="body2328348">
  <getTemperature xmlns="http://tempuri.org/temperature">
    <city xsi:Type="xsd:string">San Francisco</city>
    <state xsi:Type="xsd:string">CA</state>
    <scale xsi:Type="xsd:string">Celsius</scale>
  </getTemperature>
 </S12:Body>
</S12:Envelope>
```

10.3.2 Response

The SOAP Message Responder generates a **<SignatureResponse>** and includes it in a **<Receipt>**. Then both the <Receipt> and the **<S12:Body>** are signed together.

```
<?xml version="1.0" encoding="UTF-8"?>
<S12:Envelope xmlns:wsnr="http://schemas.reactivity.com/2003/04/wsnr/"
xmlns:S12="http://www.w3.org/2002/12/soap-envelope"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/xx/utility"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/xx/secext">
 <S12:Header>
  <wsse:Security>
   <wsse:BinarySecurityToken EncodingType="wsse:Base64Binary"
wsu:Id="ResponderCert">
MIIEfTCCA+agAwIBAgIBBDANBgkqhkiG9w0BAQQFADCB3DELMAkGA1UEBhMCVVMxEzARBgNVBAgT
CkNhbGlmb3JuaWExEDAOBgNVBAcTB0JlbG1vbnQxIDAeBgNVBAoTF1JlYWN0aXZpdHkgVGVzdCBD
b21wYW55MS4wLAYDVQQLEyVSZWFjdGl2aXR5IFRlc3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5MS4w
LAYDVQQDEyVSZWFjdGl2aXR5IFRlc3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5MSQwIgYJKoZIhvcN
AQkBFhVzb21lb25lQHNvbWV3aGVyZS5jb20wHhcNMDIwODI5MDAzMzAzWhcNMDMwODI5MDAzMzAz
WjCBmzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUx
EjAQBgNVBAoTCUNvbXBhbnkgQjEXMBUGA1UECxMOT3JnYW5pemF0aW9uIEIxEjAQBgNVBAMTCUNv
bXBhbnkgQjEkMCIGCSqGSIb3DQEJARYVc29tZW9uZUBzb21ld2hlcmUuY29tMIGfMA0GCSqGSIb3
DQEBAQUAA4GNADCBiQKBgQDc38GrOt/UYJZ8X+IIbFlaXTZiwsFYpaztru7bQrDrx9sVCd9j3q6e
xl/iILkXhQEz1tm9DEo+9VpNSTuCLhms5MHVdpFxjsJlapXyv9P4AkpyZFW/jiXx7AwP4nCtW4/6
XAOAuhQ0FJqemNUGwc5lY02lX1NxQ/gb+6qgwSOZpwIDAQABo4IBjDCCAYgwCQYDVR0TBAIwADAs
BglghkgBhvhCAQ0EHxYdT3BlblNTTCBHZW5lcmF0ZWQgQ2VydGlmaWNhdGUwHQYDVR0OBBYEFDjg
OM25FSBY3dP/9RUkIUWALqEUMIIBCgYDVR0jBIIBATCB/oAUbJviaHZTvdXnwa6fC5Df1Au+jQKh
geKkgd8wgdwxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMRAwDgYDVQQHEwdCZWxt
b250MSAwHgYDVQQKExdSZWFjdGl2aXR5IFRlc3QgQ29tcGFueTEuMCwGA1UECxMlUmVhY3Rpdml0
eSBUZXN0IENlcnRpZmljYXRlIEF1dGhvcml0eTEuMCwGA1UEAxMlUmVhY3Rpdml0eSBUZXN0IENl
cnRpZmljYXRlIEF1dGhvcml0eTEkMCIGCSqGSIb3DQEJARYVc29tZW9uZUBzb21ld2hlcmUuY29t
ggEAMAsGA1UdDwQEAwIFoDATBgNVHSUEDDAKBggrBgEFBQcDAjANBgkqhkiG9w0BAQQFAAOBgQC0
3kh6717zx047qFtrXNG3jCReJchBq/gHj5LNJoFBzqpRVtVW3i+Ef5toia2rHg1NTm3TTBPhaLTj
wUsXB9BF4K5k9qN9TuRcezk07Irtn99QNbL5xwndXEM6ZTXFpNISmmwN9Qea/H2HKRMBKH7pUJAR
nRqxPfbzODle2WH88g==</wsse:BinarySecurityToken>
    <ds:Signature xmlns:ds="http://wuw.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-
c14n-20010315"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"/>
        <ds:Reference URI="#body2328349">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>22bCiTBhM18P5rd9oAPWuDTlHM8=</ds:DigestValue>
        </ds:Reference>
        <ds:Reference URI="#receipt2328349">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>zq4duC4wvF8hYe/ACXx28jsGBHM=</ds:DigestValue>
```

```
        </ds:Reference>
      </ds:SignedInfo>

 <ds:SignatureValue>C+5+owrA/c36aUJ3gGpCOJpy93/ueFm+eTM6ePFpKT65y23qUXO0XNfF2IQ4
cS6HcUJUZvLp3ghD
fwZw4kVcgTgMWQLaEr7PwURME7ubzyxlepHDF0M4ysxEJsJ1NCzUAn8tIFXF7Ba4ganBhCaUOZm8
3GjtRRaqqmRbi4sZuyU=</ds:SignatureValue>
      <ds:KeyInfo>
        <wsse:SecurityTokenReference>
          <wsse:Reference URI="#ResponderCert"/>
        </wsse:SecurityTokenReference>
      </ds:KeyInfo>
    </ds:Signature>
    <wsnr:Receipt ReceiptFormat="signedReceipt"
wsnr:CorrelationId="urn:uuid:f81d4fde-7dec-11d0-a765-00a0c91e6bf6"
wsu:Id="receipt2328349">
      <wsnr:SignatureResponse>

 <ds:SignatureValue>aaaWCUNlYJr/saEYyCP3PBaycNWp2w9rWqPNIdVRYV8tza5okFq1yJE9kB+k
xWovVoZItAQ+y/3R
xoSsGIwfdxZ3oUPxBsVJvPOOUtrpZDVzGLT1cM2wQebcpurJZtt4yLQz6PP/cK2jcnJHUBHijmCa
wbWqZ3+V8o+6p97j+PI=</ds:SignatureValue>
        <ds:KeyInfo>
          <wsse:SecurityTokenReference>
            <wsse:Reference URI="#ResponderCert"/>
          </wsse:SecurityTokenReference>
        </ds:KeyInfo>
      </wsnr:SignatureResponse>
      <wsu:Timestamp wsu:Id="timestamp2328349">
        <wsu:Received>2003-03-12</wsu:Received>
      </wsu:Timestamp>
    </wsnr:Receipt>
  </wsse:Security>
 </S12:Header>
 <S12:Body wsu:Id="body2328349">
  <getTemperatureResponse xmlns="http://tempuri.org/temperature">
    <temperature xsi:Type="xsd:float">18.45</temperature>
  </getTemperatureResponse>
 </S12:Body>
</S12:Envelope>
```

11 References

11.1 Normative

[RFC2119]	S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.
[UUID]	M. Mealling, P. Leach, R. Salz. *A UUID URN Namespace*, http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-00.txt**,** IETF Internet-Draft, October 2002.
[SOAP11]	W3C Note, “SOAP: Simple Object Access Protocol 1.1,” 08 May 2000.
[WSS]	Web Services Security: SOAP Message Security See: Oasis Web Services Security page: http://www.oasis-open.org/committees/wss/
[XMLDSIG]	W3C Recommendation, “XML Signature Syntax and Processing,” 12 February 2002.
[XMLENC]	W3C Recommendation, “XML Encryption Syntax and Processing,” 12 December 2002.

11.2 Non-Normative

[Schneier] B. Schneier, J. Kelsey. “Cryptographic Support for Secure Logs on Untrusted Machines,” Counterpane Systems, 23 October 1999 (http://www.counterpane.com/secure-logs.pdf).

Appendix A. Revision History

Rev	Date	By Whom	What
wd-01	2003-03-05	Eric Gravengaard	Initial version
wd-02	2003-03-10	Eric Gravengaard	Updated with comments from Grant and meeting on 3/6/2003
wd-03	2003-03-12	Eric Gravengaard	More updates and example.
wd-04	2003-04-01	Eric Gravengaard	Corrections and clarifications
wd-06	2003-05-05	Eric Gravengaard	Change of name from Web Services-Non-Repudiation to Receipt Token Profile

Appendix B. Notices