SOAP Request Authentication

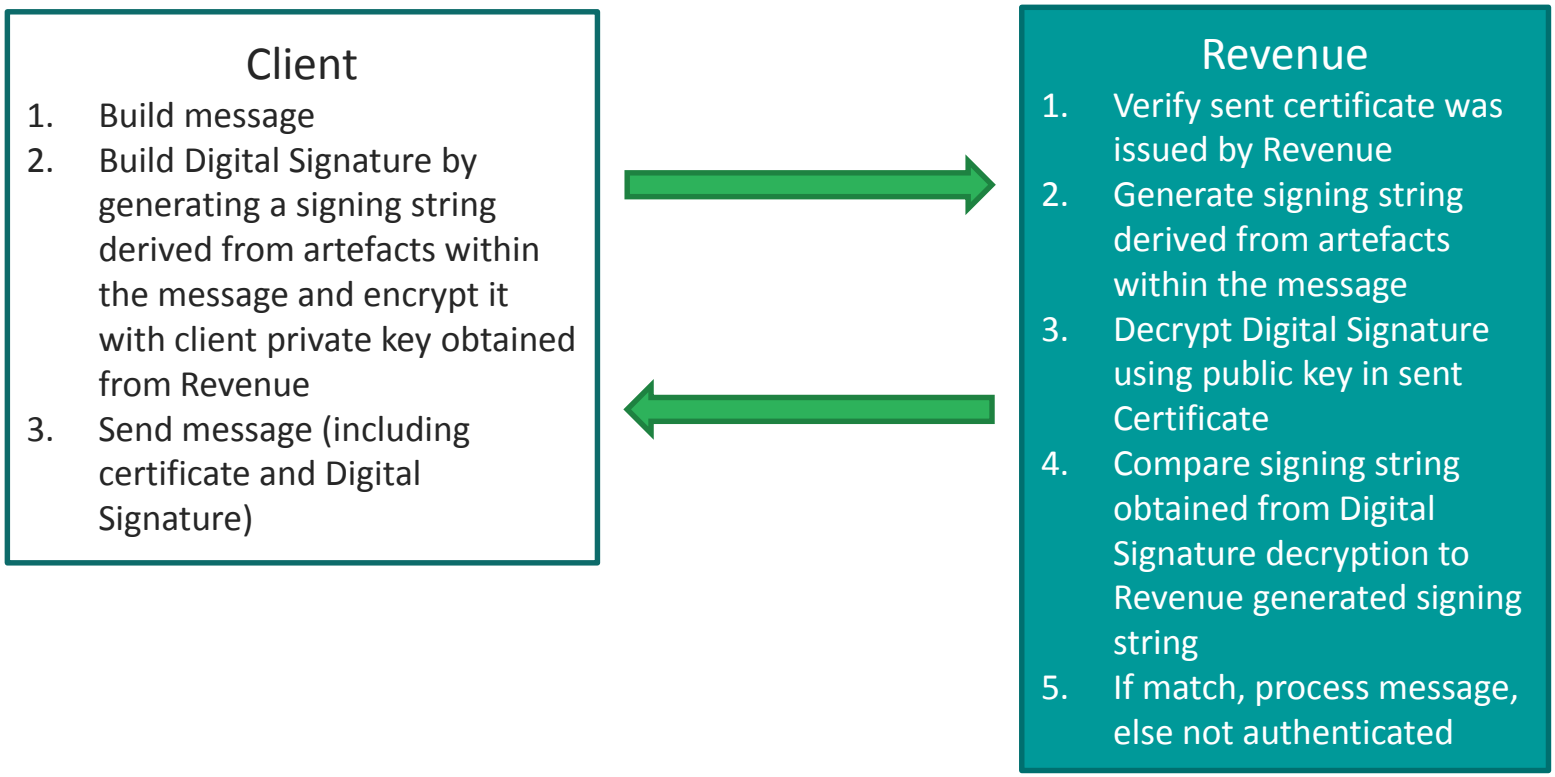
Contents of Technical Workshop

- Use of Digital Signatures
- SOAP Message Security – OASIS Standard
- Building SOAP Signature
- Questions??
- Appendix

Overview

- Any Revenue web service request that either returns confidential information or accepts submission of information must be digitally signed. This must be done using a digital certificate that has been previously retrieved from Revenue.

- The digital signature ensures the integrity of the document. By signing the document we can ensure that no malicious intruder has altered the document in any way. It is also be used for nonrepudiation purposes.



SOAP Message Security
OASIS Standard 200401, March 2004

- This OASIS specification is the result of significant work by the WSS Technical Committee .
- The goal of this specification is to enable applications to conduct secure SOAP message exchanges.
- This specification describes enhancements to SOAP messaging to provide message integrity and confidentiality. Our application only uses the specification to ensure integrity.
- The specification provides an abstract message security model in terms of security tokens combined with digital signatures to protect and authenticate SOAP messages.
- Security tokens assert claims and can be used to assert the binding between authentication secrets or keys and security identities. An authority can vouch for or endorse the claims in a security token by using its key to sign or encrypt the security token thereby enabling the authentication of the claims in the token.
- Signatures are used to verify message origin and integrity.

Key Terms

- Digest – A digest is a cryptographic checksum of an octet stream.
- 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 and/or has originated from the signer of the message, providing message integrity and authentication. The signature can be computed and verified with symmetric key algorithms, where the same key is used for signing and verifying, or with asymmetric key algorithms, where different keys are used for signing and verifying (a private and public key pair are used).
- Claim – A claim is a declaration made by an entity (e.g. name, identity, key, group, privilege, capability, etc).
- Security Token – A security token represents a collection (one or more) of claims.
- Signed Security Token – A signed security token is a security token that is asserted and cryptographically signed by a specific authority (e.g. an X.509 certificate or a Kerberos ticket)

Building a SOAP Signature Security Header

- The <wsse:Security> header block provides a mechanism for attaching security-related information targeted at a specific recipient in the form of a SOAPactor/role. This may be either the ultimate recipient of the message or an intermediary

- As elements are added to a <wsse:Security> header block, they SHOULD be prepended to the existing elements. As such, the <wsse:Security> header block represents the signing and encryption steps the message producer took to create the message. This prepending rule ensures that the receiving application can process sub-elements in the order they appear in the <wsse:Security> header block, because there will be no forward dependency among the sub-elements. Note that this specification does not impose any specific order of processing the sub-elements. The receiving application can use whatever order is required.

Building a SOAP Signature Security Header Example

```
<wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
                        <wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" wsu:Id="X509Token">MIIEmDCCA4CgAwIBAgIQWqFOg5Xcex9Y/6mNfeipuDANBgkqhBAsTCTk5OTk2MzExMDE.....iw45HSPtuyUEsfpdBqQ==</wsse:BinarySecurityToken>
                        <wsu:Timestamp wsu:Id="TS">
                                <wsu:Created>2018-09-05T13:57:41.449Z</wsu:Created>
                                <wsu:Expires>2018-09-05T13:58:11.462Z</wsu:Expires>
                        </wsu:Timestamp>
                        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                                <ds:SignedInfo>
                                        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                                        <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha512"/>
                                        <ds:Reference URI="#Body">
                                                <ds:Transforms>
                                                        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                                                </ds:Transforms>
                                                <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha512"/>

        <ds:DigestValue>+I/W7eTNgtLlOtxazFh+vjAG+MCUET2+LJ2yu3+iP4A6yJznTkdQiJ34onWE8SYAoZUYD2MKvsHt
Wz++X9wzEA==</ds:DigestValue>
                                        </ds:Reference>
                                        <ds:Reference URI="#TS">
                                                <ds:Transforms>
                                                        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                                                </ds:Transforms>
                                                <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha512"/>

        <ds:DigestValue>Jr/9zWaQ0Y7VBQE9OM6HPwzX+AD7tNnAiJFYqRdF0E+b40c9JZt/f4w0x25v9kKMszAovhgmbEQc
1ZrVzG5d2Q==</ds:DigestValue>
                                        </ds:Reference>
                                </ds:SignedInfo>
                                <ds:SignatureValue>GPBxhqQ6aBcIQDpphqboZsTJYjbM6FJcnpDckUjNymp4MZZpPpasn1eo0CWKOM3GxqKh1BiNt586
YIPrDne6gkyYtadRiLMKQKFwWpQDvjGTtjfLfi5EHzTyD2OhAH+VswOvsNraLlDC8Ph4qc6rpCu2
HwiRW/vl+8a2v3pyL/c=</ds:SignatureValue>
                                <ds:KeyInfo>
                                        <wsse:SecurityTokenReference>
                                                <wsse:Reference URI="#X509Token"/>
                                        </wsse:SecurityTokenReference>
                                </ds:KeyInfo>
                        </ds:Signature>
                </wsse:Security>
```

Building a SOAP Signature-References

- There are many motivations for referencing other message elements such as signature references or correlating signatures to security tokens. For this reason, this specification defines the wsu:Id attribute so that recipients need not understand the full schema of the message for processing of the security elements. That is, they need only "know" that the wsu:Id attribute represents a schema type of ID which is used to reference elements.

- Example:

```
<ds:Reference URI="#Body">
        <ds:Transforms>
                <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha512"/>
        <ds:DigestValue>+I/W7eTNgtLlOtxazFh+vjAG+MCUET2+LJ2yu3+iP4A6yJznTkdQiJ34onWE8SYAoZUYD2MKvsHtWz++X9
wzEA==</ds:DigestValue>
</ds:Reference>
```

Building SOAP Signature Binary Security Token

- The <wsse:BinarySecurityToken> element defines two attributes that are used to interpret it. The ValueType attribute indicates what the security token is, for example, a Kerberos ticket. The EncodingType tells how the security token is encoded, for example Base64Binary. The following is an overview of the syntax:
- Example:

<wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" wsu:Id="X509Token">MIIEmDCCA4Cg.....iw45HSPtuyUEsfpdBqQ==</wsse:BinarySecurityToken>

Questions

- ???

Appendix

- Link to the OASIS Standard documentation https://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf
- Link to gists https://gist.github.com/RevenueGitHubAdmin
- Error message guide https://revenue-ie.github.io/paye-employers-documentation/guide/Error_Structure_Guide.pdf
- Helpdesk https://revenuehelpdesk.supatools.com/login.php