Signature and Encryption Key Identifiers in Apache WSS4J

5-6 minutos

The Apache WSS4J configuration allows you to specify how to reference a public key or certificate when signing or encrypting a SOAP message via the following configuration items:

- WSHandlerConstants.SIG_KEY_ID ("signatureKeyIdentifier").
- WSHandlerConstants.ENC_KEY_ID ("encryptionKeyIdentifier").

This blog entry will explain what values are valid for each of these configuration items, and will explain what each of these values means. Firstly, let's look at what these configuration items refer to.

When creating a Signature you have the option of adding content to the Signature KeyInfo Element. This lets the recipient know what certificate/public key to use to verify the signature. Specifying a value for WSHandlerConstants.SIG_KEY_ID allows you to change how to refer to the key.

When encrypting some part of the message, a session key is typically generated and used to encrypt the message part, which is then wrapped in an EncryptedData Element. This refers (typically via a Direct Reference) to a EncryptedKey Element in the security

header, where the session key is encrypted using the public key of the recipient. Specifying a value for WSHandlerConstants.ENC_KEY_ID allows you to change how to refer to the public key of the recipient in the EncryptedKey KeyInfo element.

The following valid values for these configuration items are:

- IssuerSerial (default)
- DirectReference
- X509KeyIdentifier
- Thumbprint
- SKIKeyIdentifier
- KeyValue (signature only)
- EncryptedKeySHA1 (encryption only)

1) IssuerSerial

This (default) key identifier method means that the Issuer Name and Serial Number of a X.509 Certificate is included directly in the KeyInfo Element. For example:

```
<ds:KeyInfo>
   <wsse:SecurityTokenReference>
      <ds:X509Data>
         <ds:X509IssuerSerial>

<ds:X509IssuerName>CN=XYZ</ds:X509IssuerName>
            <ds:X509SerialNumber>124124....
</ds:X509SerialNumber>
         </ds:X509IssuerSerial>
```

```
        </ds:X509Data>
    </wsse:SecurityTokenReference>
</ds:KeyInfo>
```

The certificate is not included in the message and so the recipient will have to have access to the certificate matching the given Issuer Name and Serial Number in a keystore.

2) DirectReference

This key identifier method is used when the X.509 Certificate is included in the message, unlike the IssuerSerial case above. The certificate is Base-64 encoded and included in the request via a BinarySecurityToken element, e.g.:

```
<wsse:BinarySecurityToken EncodingType="...#Base64Binary"
    ValueType="...#X509v3">MIIBY...
</wsse:BinarySecurityToken>
```

This Certificate is then referred to directly from the KeyInfo Element as follows:

```
<ds:KeyInfo>
    <wsse:SecurityTokenReference>
        <wsse:Reference URI="#X509-..."
ValueType="...#X509v3"/>
    </wsse:SecurityTokenReference>
</ds:KeyInfo>
```

3) X509KeyIdentifier

This key identifier method is similar to DirectReference, in that the certificate is included in the request. However, instead of referring to a certificate, the certificate is included directly in the KeyInfo element, e.g.:

```
<ds:KeyInfo>
   <wsse:SecurityTokenReference>
      <wsse:KeyIdentifier EncodingType="...#Base64Binary"
         ValueType="...#X509v3">MIIB...
      </wsse:KeyIdentifier>
   </wsse:SecurityTokenReference>
</ds:KeyInfo>
```

4) Thumbprint

This Key Identifier method refers to the Certificate via a SHA-1 Thumbprint. The certificate may or may not be included in the request. For example:

```
<ds:KeyInfo>
   <wsse:SecurityTokenReference>
      <wsse:KeyIdentifier EncodingType="...#Base64Binary"
         ValueType="...#ThumbprintSHA1">
         5epW9GhL6s0kC9X9egsRZ90ooeE=
      </wsse:KeyIdentifier>
   </wsse:SecurityTokenReference>
</ds:KeyInfo>
```

5) SKIKeyIdentifier

This Key Identifier method refers to a Certificate via a Base-64 encoding of the Subject Key Identifier, e.g.:

```
<ds:KeyInfo>
   <wsse:SecurityTokenReference>
      <wsse:KeyIdentifier EncodingType="...#Base64Binary"
         ValueType="...#X509SubjectKeyIdentifier">
         2DUoN4ppxJz/RNgcCDsJ4SocPdk=
```

```
        </wsse:KeyIdentifier>
    </wsse:SecurityTokenReference>
</ds:KeyInfo>
```

6) KeyValue

This Key Identifier method only applies for Signatures. It includes the (RSA) PublicKey directly in the Signature KeyInfo Element as follows:

```
<ds:KeyInfo>
    <ds:KeyValue>
        <ds:RSAKeyValue>
            <ds:Modulus>tfJ29N0G1...</ds:Modulus>
            <ds:Exponent>AQAB</ds:Exponent>
        </ds:RSAKeyValue>
    </ds:KeyValue>
</ds:KeyInfo>
```

7) EncryptedKeySHA1

This Key Identifier method only applies for Encryption. Unlike the previous methods it refers to the way the EncryptedData references the EncryptedKey Element, rather than the way the EncryptedKey Element refers to the public key. For example:

```
<ds:KeyInfo>
    <wsse:KeyIdentifier EncodingType="...#Base64Binary"
        ValueType="...#EncryptedKeySHA1">
        X/8wvCY...
    </wsse:KeyIdentifier>
</ds:KeyInfo>
```