Add the following section after section 5.3.10:

5.3.11 KeyValueToken Assertion

This element represents a requirement for a KeyValue token. The next section defines the KeyValue security token abstraction for purposes of this token assertion.

This document defines requirements for KeyValue token when used in combination with RSA cryptographic algorithm. Additional cryptographic algorithms can be introduced by another specifications by introducing new nested assertions besides *sp:RsaKeyValue*.

Syntax

```
<sp:KeyValueToken sp:IncludeToken=”xs:anyURI”? xmlns:sp=”...” ... >
  <wsp:Policy xmlns:wsp=”...”>
    <sp:RsaKeyValue ... /> ?
    ...
  </wsp:Policy> ?
  ...
</sp:RsaToken>
```

The following describes the attributes listed in the schema outlined above:

/sp:KeyValueToken
: This identifies a RsaToken assertion.

/sp:KeyValueToken/@sp:IncludeToken
: This optional attribute identifies the token inclusion value for this token assertion.

/sp:KeyValueToken/wsp:Policy
: This optional element identifies additional requirements for use of the sp:KeyValueToken assertion.

/sp:KeyValueToken/wsp:Policy/sp:RsaKeyValue
: This optional element is a policy assertion that indicates that the *ds:RSAKeyValue* element must be present in the KeyValue token. This indicates that an RSA key pair must be used.

5.3.11.1 KeyValue Token

XML Signature specification allows reference an arbitrary key pair by using the corresponding public key value. This allows using an arbitrary key pair to sign or encrypt XML elements. The purpose of this section is to define the KeyValue token abstraction that represents such key pair referencing mechanism.

Although the *ds:KeyValue* element as defined in the XML Signature specification is generic enough to be used with any asymmetric cryptographic algorithm this document only profiles the usage of *ds:KeyValue* element in combination with RSA cryptographic algorithm.

The RSA key pair is represented by the *ds:KeyInfo* element containing the *ds:KeyValue* element with the RSA public key value in *ds:RSAKeyValue* as defined in the XML Signature specification:

```
<ds:KeyInfo xmlns=”http://www.w3/org/2000/09/xmldsig#”>
  <ds:KeyValue>
    <ds:RSAKeyValue>
      <ds:Modulus>ds:CryptoBinary</ds:Modulus>
      <ds:Exponent>ds:CryptoBinary</ds:Exponent>
    </ds:RSAKeyValue>
  <ds:KeyValue>
</ds:KeyInfo>
```

When the KeyValue token is used the corresponding public key value appears directly in the signature or encrypted data *ds:KeyInfo* element like in the following example. There is no KeyValue token manifestation outside the *ds:KeyInfo* element.

```
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" />
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
    <Reference URI="#_1">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue>...</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>...</SignatureValue>
  <KeyInfo>
    <KeyValue>
      <RSAKeyValue>
        <Modulus>...</Modulus>
        <Exponent>...</Exponent>
      </RSAKeyValue>
    </KeyValue>
  </KeyInfo>
</Signature>
```

Since there is no representation of the KeyValue token outside the *ds:KeyInfo* element and thus no identifier can be associated with the token, the KeyValue token cannot be referenced by using *wsse:SecurityTokenReference* element. However the *ds:KeyInfo* element representing the KeyValue token can be used whenever a security token can be used as illustrated on the following example:

```
<t:RequestSecurityToken xmlns:t="...">
  <t:RequestType>...</t:RequestType>
  ...
  <t:UseKey>
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <KeyValue>
        <RSAKeyValue>
          <Modulus>...</Modulus>
          <Exponent>...</Exponent>
        </RSAKeyValue>
```

```
      </KeyValue>
    </KeyInfo>
  </t:UseKey>
</t:RequestSecurityToken>
```