In the previous sections, we considered two security policy assertions. In this section, let us look at one of the security policy assertions in a little more detail.

As you would expect, securing messages can be a complex usage scenario. If Company-X uses the sp:TransportBinding policy assertion to indicate the use of transport-level security for protecting messages just indicating the use of transport-level security for protecting messages may not be sufficient. To successfully interact with Company-X's Web services, the developer must also know what transport token to use, what particular secure transport to use, what specific algorithm suite to use for performing cryptographic operations, etc. The sp:TransportBinding policy assertion can represent these dependent behaviors. In this section, let us look at how to capture these dependent behaviors in a machine-readable form.

A policy assertion – like the sp:TransportBinding - identifies a visible domain specific behavior that is a requirement. Given an assertion, there may be other dependent behaviors that need to be enumerated for a Web Service interaction. In the case of the sp:TransportBinding policy assertion, Company-X needs to identify the use of a transport token, a secure transport, an algorithm suite for performing cryptographic operations, etc. A nested policy expression can be used to enumerate such dependent behaviors.

What is a nested policy expression? A nested policy expression is a policy expression that is a child element of a parent policy assertion element. A nested policy expression further qualifies the behavior of its parent policy assertion. The qualification may indicate a relationship or context between the parent policy assertion and a nested policy expression [link to section 3.1 in Framework].

In the example below, the child Policy element is a nested policy expression and further qualifies the behavior of the sp:TransportBinding policy assertion within the security domain. The sp:TransportToken is a nested policy assertion of the sp:TransportBinding policy assertion. The sp:TransportToken assertion requires the use of a specific transport token and further qualifies the behavior of the sp:TransportBinding policy assertion (which already requires the use of transport-level security for protecting messages).

Example 2-13. Transport Security Policy Assertion

```
<sp:TransportBinding>
  <Policy>
    <sp:TransportToken>
      <Policy>
```

Formatted: Highlight
Deleted: is
Formatted: Highlight
Formatted: Highlight
Deleted: . J
Formatted: Highlight
Deleted: is
Formatted: Highlight
Deleted: not
Formatted: Highlight
Formatted: Highlight
Formatted: Highlight
Formatted: Highlight
Formatted: Highlight
Formatted: Highlight
Formatted: Highlight
Formatted: Highlight
Formatted: Highlight
Formatted: Right: 18 pt

```
      <sp:HttpsToken>
        <wsp:Policy/>
      </sp:HttpsToken>
    </Policy>
  </sp:TransportToken>
  <sp:AlgorithmSuite>
    <Policy>
      <sp:Basic256Rsa15/>
    </Policy>
  </sp:AlgorithmSuite>
  …
 </Policy>
</sp:TransportBinding>
```

The sp:AlgorithmSuite is a nested policy assertion of the sp:TransportBinding policy assertion. The sp:AlgorithmSuite assertion requires the use of the algorithm suite identified by its nested policy assertion (sp:Basic256Rsa15 in the example above) and further qualifies the behavior of the sp:TransportBinding policy assertion.

Setting aside the details of using transport-level security, Web service developers can use a policy-aware client that recognizes this policy assertion and engages transport-level security and its dependent behaviors automatically. That is, the complexity of security usage is absorbed by a policy-aware client and hidden from these Web service developers.

In another example, WS-Security Policy defines a sp:HttpToken assertion to contain three possible nested elements, sp:HttpBasicAuthentication, sp:HttpDigestAuthentication and sp:RequireClientCertificate. When the HttpToken is used with an empty nested policy in a policy expression by a provider, it will indicate that none of the dependent behaviors namely authentication or client certificate is required. A non-anonymous client who requires authentication or client certificate will not be able to use this provider solely on the basis of Framework intersection algorithm alone.

Formatted: Highlight

Formatted: Right: 18 pt

Example 2-14. Empty Nested Assertion

```
<sp:TransportToken>
   <wsp:Policy>
     <sp:HttpsToken>
      <wsp:Policy/>
     </sp:HttpsToken>
   </wsp:Policy>
</sp:TransportToken>
```

…

3.3 Policy Data Model

...considered nested policy expressions in the context of a security usage scenario. Let us look at its shape in the policy data model. In the normal form, a nested policy is a policy that has at most one policy alternative and is related to or has a context when it exists in its parent policy assertion. The policy alternative in a nested policy represents a collection of associated or dependent behaviors or requirements or conditions that qualify the behavior of its parent policy assertion….

Formatted: Highlight

Deleted: owned by

Formatted: Highlight

3.4 Compatible Policies

For this interaction, the developer's policy-aware client can use policy alternative (a) to satisfy Company-X's conditions or requirements.

Similarly, policy intersection can be used to check if providers expose endpoints that conform to a standard policy. For example, a major retailer might require all their supplier endpoints to be compatible with an agreed upon policy.

Policy assertions and nested policy expressions are evaluated in context for compatibility matching during intersection processing. Where they exist, nested policy expressions are evaluated in the context of their parent policy assertions relative to the policy alternatives.

In compatibility matching, a nested policy expression has a different context if it exists in two different places and is related to two QNames within two policy alternatives evaluated.

Formatted: Right: 18 pt

NOTE: An example may be required here and we have one if needed.

Formatted: Font: Bold, Highlight
Formatted: Highlight
Formatted: Font: Bold

Formatted: Right: 18 pt