PLEASE DO NOT REPLY TO THIS EMAIL OR START A DISCUSSISON THREAD UNTIL THE ISSUE IS ASSIGNED A NUMBER.

The issues coordinators will notify the list when that has occurred.

Protocol: ws-trust

http://www.oasis-open.org/apps/org/workgroup/ws-sx/download.php/17403/ws-trust-1.3-spec-ed-01-r05-diff.pdf

Artifact: spec

Type: design

Title: Key and Encryption Requirements Clarification

Description:
Currently, the section 9.2 in WS-Trust specification defines various elements that can be used to request specific types of keys or algorithms for the returned token and the RSTRC message. The current description of some of those parameters is not clear enough for the reader to understand how they affect the issued token and the returned RSTRC message. This issue proposes an additional description to clarify those parameters.
This issue already depends on the other issue that is being raised that proposes to add a new optional parameter to the RST – KeyWrapAlgorithm.

Related issues:
NEW Issue: Missing KeyWrapAlgorithm requirement in section 9.2

Proposed Resolution:
Add the following text after the line 2002:

The following table summarizes the various algorithm parameters defined above. ***T*** is the issued token, ***P*** is the proof key.

SignatureAlgorithm - The signature algorithm to use to sign ***T***

EncryptionAlgorithm - The encryption algorithm to use to encrypt ***T***

CanonicalizationAlgorithm - The canonicalization algorithm to use when signing ***T***

ComputedKeyAlgorithm - The key derivation algorithm to use if using a symmetric key for ***P*** where ***P*** is computed using client, server, or combined entropy.

Encryption - The token/key to use when encrypting ***T***

ProofEncryption - The token/key to use when encrypting ***P***

UseKey - This is ***P***. This is generally used when the client supplies a public-key that it wishes to be embedded in ***T*** as the proof key.

SignWith - The signature algorithm the client intends to employ when using ***P*** to sign.

EncryptWith - The encryption algorithm the client intends to employ when using ***P*** to encrypt.

The encryption algorithms further differ based on whether the issued token contains asymmetric key or symmetric key. Furthermore, they differ based on what type of key is used to protect the issued token from the STS to the relying party. The following cases can occur:

***T* contains symmetric key/STS uses symmetric key to encrypt *T* for RP**
EncryptWith – used to indicate symmetric algorithm that client will use to protect message to RP when using the proof key (e.g. AES256)
EncryptionAlgorithm – used to indicate the symmetric algorithm that the STS should use to encrypt the ***T*** (e.g. AES256)

***T* contains symmetric key/STS uses asymmetric key to encrypt *T* for RP**
EncryptWith – used to indicate symmetric algorithm that client will use to protect message to RP when using the proof key (e.g. AES256)
EncryptionAlgorithm – used to indicate the symmetric algorithm that the STS should use to encrypt ***T*** for RP (e.g. AES256)
KeyWrapAlgorithm – used to indicate the KeyWrap algorithm that the STS should use to wrap the generated key that is used to encrypt the ***T*** for RP.

***T* contains asymmetric key/STS uses symmetric key to encrypt *T* for RP**
EncryptWith – used to indicate the KeyWrap algorithm that the client will use to protect symmetric key that is used to protect message to RP when using the proof key (e.g. RSA-OAEP-MGF1P)
EncryptionAlgorithm – used to indicate the symmetric algorithm that the STS should use to encrypt ***T*** for RP (e.g. AES256)

***T* contains asymmetric key/STS uses asymmetric key to encrypt *T* for RP**
EncryptWith - used to indicate the KeyWrap algorithm that the client will use to protect symmetric key that is used to protect message to RP when using the proof key (e.g. RSA-OAEP-MGF1P)
EncryptionAlgorithm - used to indicate the symmetric algorithm that the STS should use to encrypt ***T*** for RP (e.g. AES256)
KeyWrapAlgorithm – used to indicate the KeyWrap algorithm that the STS should use to wrap the generated key that is used to encrypt the ***T*** for RP.