NCIA Registry
04 FEB 2015
The Hague

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

NATO Communications and Information Agency
Agence OTAN d'information et de communication

AGENCY INSTRUCTION

INSTR TECH 06.02.03

Service Interface Profile for Security Token Services

Effective date:

Revision No: Original

Issued by: Chief, Core Enterprise Services

Approved by: Director, Service Strategy

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

Table of Amendments

Amendment No	Date issued	Remarks

Author Details

Organization	Name	Contact Email/Phone
NCI Agency	R. Fiske	rui.fiske@ncia.nato.int
NCI Agency	M. Lehmann	marek.lehmann@ncia.nato.int
NCI Agency	R. Malewicz	robert.malwicz@ncia.nato.int
NCI Agency	L. Schenkels	leon.schenkels@ncia.nato.int
NCI Agency	D. Gujral	davinder.gujral@ncia.nato.int

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

Table of Contents

List of Annexes

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

AGENCY INSTRUCTION 06.02.03

Service Interface Profile for Security Token Services

0 PRELIMINARY INFORMATION

0.1 References

A. NCIA/GM/2012/235; Directive 1 Revision 1; dated 3 May 2013
B. NCIARECCEN-4-22852 DIRECTIVE 01.01 Agency Policy on Management and Control of Directives, Notices, Processes, Procedures and Instructions, dated 20 May 2014
C. NCIARECCEN-4-23297, Directive 06.00.01, Management and Control of Directives, Processes, Procedures and Instructions on Service Management, dated 03 June 2014

0.2 Purpose

This Technical Instruction (TI) provides detailed information, guidance, instructions, standards and criteria to be used when planning, programming, and designing Agency products and services. In this specific case the TI defines a Service Interface Profile (SIP) for one of NATO's Core Enterprise Services.

TIs are living documents and will be periodically reviewed, updated, and made available to Agency staff as part of the Service Strategy responsibility as Design Authority. Technical content of these instructions is the shared responsibility of SStrat/Service Engineering and Architecture Branch and the Service Line of the discipline involved.

TIs are primarily disseminated electronically[1], and will be announced through Agency Routine Orders. Hard copies or local electronic copies should be checked against the current electronic version prior to use to assure that the latest instructions are used.

0.3 Applicability

This TI applies to all elements of the Agency, in particular to all NCI Agency staff involved in development of IT services or software products. It is the responsibility of all NCI Agency Programme, Service, Product and Project Managers to ensure the implementation of this technical instruction and to incorporate its content into relevant contractual documentation for external suppliers.

1 SIP INTRODUCTION

One of the key components of the NNEC security infrastructure identified in [NCIA TR/2012/CPW007253/30, 2012], which must be read along with this Technical Report, is the *Security Token Service (STS)*. This service is responsible for issuing extensible markup language (XML) *Security Tokens* to entities within the enterprise that can be used for the purpose of *Authentication* to a service. The service can then make *Authorization* decisions based on the set of *Attributes*, or *Claims*, that are contained within the *Security Token*, either internally or through reference to a *Policy Decision Point (PDP)*. Tokens can be issued to *Active Clients* (i.e. those that are able to make simple object access protocol (SOAP) calls directly) or *Passive Clients*, which cannot make SOAP calls (the most common example of which is a web browser).

This specification provides the interface control for the *STS* for both *Active* and *Passive Clients*. It specifies the structure of messages that are sent to the *STS*, and the response that is returned from it. This profile has evolved in response to the available technologies and mechanisms that can be used to apply security within a service-oriented environment.

[1] https://servicestrategy.nr.ncia/SitePages/Agency%20Directives%20(Technical).aspx

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

The purpose of this Service Interface Profile (SIP) is to specify how the security token service component of the Core Enterprise Services (CES) Security Services may be called. This covers only the call from a client to the *STS*, and the corresponding response. This includes how the *Message* must be structured and the elements that must be contained within the call. It does not cover what happens after the token has been retrieved (in other words, how a protected service will be called) or what happens at the *Authorization* stage of the process (contact with the *PDP*). These will be covered by [NCIA TR/2012/CPW007253/02, 2012], [NCIA TR/2012/CPW007253/06, 2012].

1.1 Terminology

For the terminology used in this document, please see [NCIA TR/2012/CPW007253/02, 2012].

1.2 Relationships to other profiles and specifications

1.2.1 Normative references

In addition to the normative references in [NCIA TR/2012/CPW007253/02, 2012], the following documents have fed into this specification, and are incorporated as normative references:

1.2.1.1 WS-Trust 1.4

http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.doc

1.2.1.2 WS-Federation 1.1

http://specs.xmlsoap.org/ws/2006/12/federation/ws-federation.pdf

1.2.1.3 WS-Federation: Passive Requestor Profile

http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-fedpass/ws-fedpass.pdf

1.2.1.4 Web Services Security: SOAP Message Security 1.1 (OASIS)

http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

2 SIP DEFINITION

2.1 Subject

In order to access resources protected by the NNEC CES Security Services, the service consumer must present an XML *Security Token* for *Authentication*, that is, to present the credentials of the consuming entity to the service provider. Within the infrastructure of the CES, this token must be issued by a trusted party, or *Identity Provider (IdP)*. For a more detailed description of the mechanisms used to protect services, see [NC3A RD-2814, 2009].

Thus, any service consumer must retrieve a token from an *IdP* prior to calling a service. This SIP defines the mechanisms that can be used to retrieve the token. It does not cover how the token is used, nor some of the specific contents of the token, such as the claims included within it, nor the lifetime values of the token.

This SIP also does not cover the establishment of cross-domain federation of identity. However, it does cover delegation, where one service consumes data from another service on behalf of the original calling consumer. It also covers both *Active* and *Passive Clients*.

2.1.1 High-level View

Figure 1 shows the high-level message exchange when retrieving a *Security Token*.

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

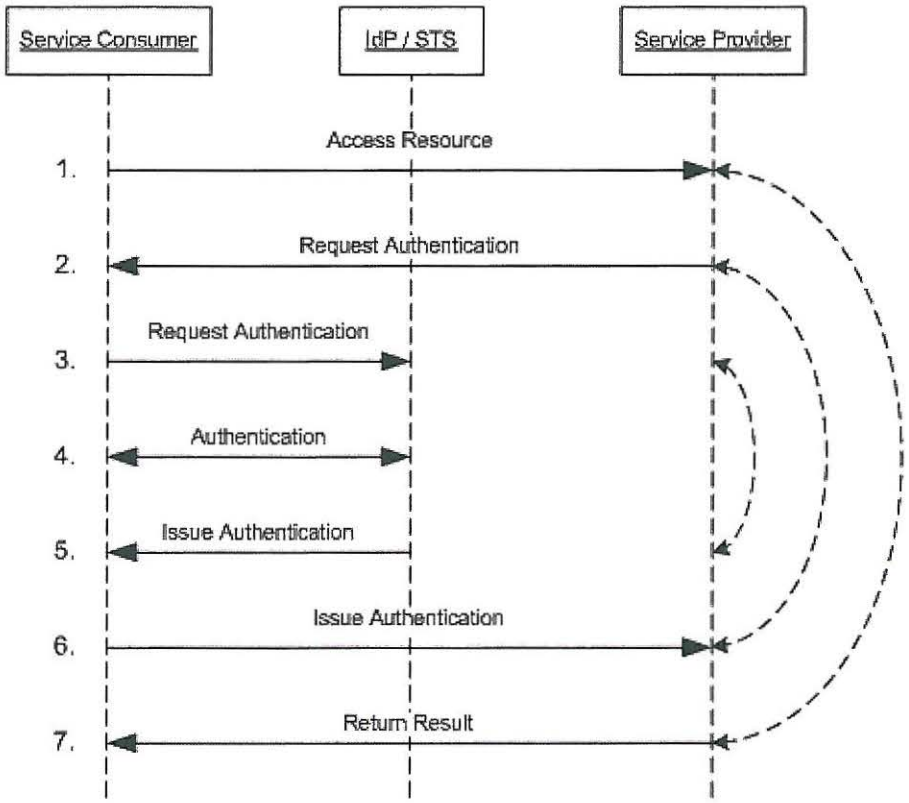

Figure 1 High-level message exchange (source: [WS-Federation, 2006])

Steps 1 and 2, where the client contacts the resource provider prior to requesting the *Security Token,* are OPTIONAL. The consumer MAY request the token for inclusion in the call prior to contacting the resource provider.

2.1.2 WS-Trust and WS-Federation

The primary protocol for requesting *Security Tokens* is [OASIS WS-Trust, 2009]. The reference version of this specification for this SIP is 1.4.

However, since [OASIS WS-Trust, 2009] specifies the message exchanges through a web service, this is not suitable for *Passive Clients,* which – by definition – are unable to issue web service calls. As described in [WS-Federation, 2006]:

> "The primary issue for *Web browsers* is that there is no easy way to directly issue SOAP requests. Consequently, processing must be performed within the confines of the base HTTP 1.1 functionality (GET, POST, redirects, and cookies) and conform as closely as possible to the [OASIS WS-Trust, 2009] protocols for token acquisition."

Therefore, this SIP also includes the use of [WS-Federation, 2006] 1.1, and specifically the [WS-Federation: Passive Requestor Profile, 2003], [WS-Federation, 2006] are an extension to [OASIS WS-Trust, 2009]. Therefore the two protocols share a number of data structures.

The *Security Token Service* MUST support the mandatory components of both [OASIS WS-Trust, 2009] 1.4 and [WS-Federation, 2006] 1.1.

A client MUST support either [OASIS WS-Trust, 2009] 1.4 or [WS-Federation, 2006] 1.1. A client MAY support both protocols.

A service, which provides no user interface (UI) for a user, MUST support [OASIS WS-Trust, 2009] 1.4.

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

2.1.3 Authentication

In order for the *STS* to issue a token to a requestor, whether using [OASIS WS-Trust, 2009] or [WS-Federation, 2006], the *STS* MUST authenticate the credentials of the user in order to establish their identity. This SIP does not prohibit any *Authentication* mechanisms that are permitted by the normative specifications, however the following hypertext transfer protocol (HTTP) mechanisms MUST be supported as a minimum for both [OASIS WS-Trust, 2009] and [WS-Federation, 2006] end-points:

- Kerberos ([IETF RFC 4120, 2005])
- X.509 Client Certificates ([IETF RFC 2246, 1999]).

2.2 Service Interface

Two interfaces MUST be presented by the *STS*, one for [OASIS WS-Trust, 2009] (as a web service) and one for [WS-Federation, 2006] (as a web application). Thus, the SIP contains a set of operations, inputs, outputs, and errors for both these standards. The type of input will depend on whether the caller is an *Active* or *Passive Client*. *Passive Clients* can only use [WS-Federation, 2006].

3 WS-TRUST

3.1 Operations

The operations that are specified here are the minimal operations that MUST be implemented by the *STS* in order to support the issuance of tokens. Other operations that are defined by the relevant specification MAY be implemented by the *STS* in accordance with those specifications.

3.1.1 Operation: Issue

Based on the credential provided/proven in the request, a new token is issued, possibly with new proof information.

3.1.1.1 WS-Addressing Actions

- `http://docs.oasis-open.org/ws-sx/ws-trust/200512/RST/Issue`
- `http://docs.oasis-open.org/ws-sx/ws-trust/200512/RSTR/Issue`
- `http://docs.oasis-open.org/ws-sx/ws-trust/200512/RSTRC/IssueFinal`.

For this operation, the `wst:RequestType` element (as described in Section3.2.8) MUST use the following uniform resource identifier (URI):

- `http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue`.

3.1.2 Operation: Renew

A previously issued token with expiration is presented (and possibly proven) and the same token is returned with new expiration semantics.

3.1.2.1 WS-Addressing Actions

- `http://docs.oasis-open.org/ws-sx/ws-trust/200512/RST/Renew`
- `http://docs.oasis-open.org/ws-sx/ws-trust/200512/RSTR/Renew`
- `http://docs.oasis-open.org/ws-sx/ws-trust/200512/RSTR/RenewFinal`.

For this operation, the `wst:RequestType` element (as described in Section3.2.8) MUST use the following URI:

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

- `http://docs.oasis-open.org/ws-sx/ws-trust/200512/Renew`

3.2 Inputs

Active Clients using [OASIS WS-Trust, 2009] MUST wrap their *Message* in a `soap:Envelope` element.

The requestor of the token, MUST format the *Message* with a [W3C WS-Addressing, 2006] *Header* with the appropriate `wsa:Action`, as specified in section 3.1 of this document. The `soap:Body` element MUST contain a `RequestSecurityTokenCollection` or `RequestSecurityToken` element.

Requestors MUST NOT request a token anonymously. The *STS* MAY reject or drop anonymous requests.

3.2.1 SOAP

An [OASIS WS-Trust, 2009] request MUST be submitted in a SOAP envelope with a SOAP *Header* element. The version of SOAP to use is not specified by this SIP, but is specified in [NCIA TR/2012/SPW008000/30, 2012].

3.2.2 WS-Addressing Info

[W3C WS-Addressing, 2006] contains metadata about the *Message*. For further information about the use of [W3C WS-Addressing, 2006] in message exchanges, see [NCIA TR/2012/SPW008000/30, 2012].

3.2.2.1 Element(s)

Element	Notes
`/soap:Envelope/soap:Header/wsa:Action`	This is REQUIRED. It MUST use one of the URIs included in Section 3.1.
`/soap:Envelope/soap:Header/wsa:MessageID`	This is REQUIRED and MUST be unique for each call to the service.
`/soap:Envelope/soap:Header/wsa:ReplyTo`	This is OPTIONAL with the default value of: `http://www.w3.org/2005/08/addressing/anonymous`. This specifies the end-point reference for the intended receiver for replies to this message. A caller MAY explicitly use the anonymous value, `http://www.w3.org/2005/08/addressing/anonymous`.
`/soap:Envelope/soap:Header/wsa:To`	This is REQUIRED, and is the end-point of the *STS* service.

3.2.3 SOAP Body

3.2.3.1 Element(s)

Element	Notes
`/soap:Envelope/soap:Body`	This contains the actual request for the *Security Token.*

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

3.2.4 RequestSecurityTokenCollection

3.2.4.1 Element(s)

Element	Notes
`/soap:Envelope/soap:Body/wst:RequestSecurityTokenCollection`	This is the structure for requesting multiple tokens in one request. It is OPTIONAL. However, if it is present it MUST contain at least one `wst:RequestSecurityToken` element.

3.2.5 RequestSecurityToken

As highlighted in Section 3.2.4, this may be included as part of a request for multiple *Security Tokens*, as part of a `wst:RequestSecurityTokenCollection` element. However, for the sake of simplicity the following structures assume that it is a direct child of the `soap:Body`.

3.2.5.1 Element(s)

Element	Notes
`/soap:Envelope/soap:Body/wst:RequestSecurityToken`	This is the structure that contains the request to the *STS*, and is therefore REQUIRED.

3.2.6 AppliesTo

This specifies the scope of the requested token. If a token issuer cannot provide a token with a scope that is at least as broad as that requested by the requestor then it SHOULD generate a fault. As specified in Section 4.4.1 of [OASIS WS-Trust, 2009], "The requestor and issuer MUST agree on the version of [WS-Policy] used to specify the scope of the issued token."[2]

3.2.6.1 Element(s)

Element	Notes
`/soap:Envelope/soap:Body/wst:RequestSecurityToken/wsp:AppliesTo`	This REQUIRED element specifies the scope of the *Relying Party*.

3.2.7 TokenType

3.2.7.1 Element(s)

Element	Notes
`/soap:Envelope/soap:Body/wst:RequestSecurityToken/wst:TokenType`	In this SIP, this element is REQUIRED. The value of the TokenType MUST be a SAML 2.0 token: `http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0`

3.2.8 RequestType

[2] [W3C WS-Policy 1.5, 2007]

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

3.2.8.1 Element(s)

Element	Notes
`/soap:Envelope/soap:Body/wst:RequestSecurityToken/wst:RequestType`	This is REQUIRED, and MUST have the appropriate value for the operation as listed in Section 3.1. Thus, for an issue request, it would have a value of: `http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue`

3.2.9 ActAs

This element allows the requestor to request a delegated token from the *STS*, so that the identity of the original caller can be maintained throughout the calling chain.

- It is OPTIONAL when requesting a normal *Security Token*.
- It is MANDATORY when requesting a delegated *Security Token*.

Note: `ActAs` was introduced in the [OASIS WS-Trust, 2009] 1.4 specification. In [OASIS WS-Trust, 2009], the `ActAs` element is specified as an element from the [OASIS WS-Trust, 2009] 1.3 namespace. However, this is in contradiction to other normative statements in [OASIS WS-Trust, 2009] and has thus been submitted to the OASIS technical committee as an error. Most implementations take the following approach, which is incorporated here:

- The `ActAs` element MUST come under the [OASIS WS-Trust, 2009] 1.4 namespace.
- The `ActAs` element MUST contain a *Security Token* that can be verified and validated by the *STS*.
- The *Security Token* contained within the `ActAs` element MUST be a security assertion markup language (SAML) token.
- The *Security Token* contained within the `ActAs` element SHOULD be a SAML 2.0 token.

3.2.9.1 Element(s)

Element	Notes
`/wst:RequestSecurityToken/wst14:ActAs`	Contains the *Security Token* for which the delegated token is to be issued.

3.3 Outputs

The output of the *STS* is a SAML 2.0 assertion, which MUST have the structure described in [NCIA TR/2012/CPW007253/02, 2012].

The response to the *Security Token* response to a [OASIS WS-Trust, 2009] request MUST be returned in a SOAP envelope.

The structure of the message is as follows:

3.3.1 SOAP

An [OASIS WS-Trust, 2009] response MUST be submitted in a SOAP envelope with a SOAP *Header* element. The version of SOAP to use is not specified by this SIP, but is specified in [NCIA TR/2012/SPW008000/30, 2012].

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

3.3.2 WS-Addressing Info

[W3C WS-Addressing, 2006] contains metadata about the message.

3.3.2.1 Element(s)

Element	Notes
`/soap:Envelope/soap:Header/wsa:Action`	This is REQUIRED, as it specifies what action the *Message* is in response to.
`/soap:Envelope/soap:Header/wsa:RelatesTo`	This is REQUIRED and MUST relate to the corresponding `MessageID` in the request *Message*.

3.3.3 SOAP Body

3.3.3.1 Element(s)

Element	Notes
`/soap:Envelope/soap:Body`	This contains the response to the request for the *Security Token*.

3.3.4 RequestSecurityTokenResponseCollection

This is the structure that contains the individual *Security Token* responses, and therefore MUST be present.

- It MUST contain at least one `RequestSecurityTokenResponse` element.
- It MAY contain more than one `RequestSecurityTokenResponse` element.

3.3.4.1 Element(s)

Element	Notes
`/soap:Envelope/soap:Body/wst:RequestSecurityTokenResponseCollection`	This is REQUIRED.

3.3.5 RequestSecurityTokenResponse

3.3.5.1 Element(s)

Element	Notes
`/soap:Envelope/soap:Body/wst:RequestSecurityTokenResponseCollection/wst:RequestSecurityTokenResponse`	This is REQUIRED.

3.3.6 AppliesTo

3.3.6.1 Element(s)

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

Element	Notes
`/soap:Envelope/soap:Body/wst:RequestSecurityTokenResponseCollection/wst:RequestSecurityTokenResponse/wsp:AppliesTo`	This RECOMMENDED element specifies the scope of the *Relying Party* for which the token is valid.

3.3.7 RequestedSecurityToken

3.3.7.1 Element(s)

Element	Notes
`/soap:Envelope/soap:Body/wst:RequestSecurityTokenResponseCollection/wst:RequestSecurityTokenResponse/wst:RequestedSecurityToken`	This contains the SAML token, as described in [NCIA TR/2012/CPW007253/02, 2012], and so is REQUIRED.

3.4 Errors

There are many security-related errors that may be raised as SOAP faults by the *STS*. The actual errors will depend on the implementation of the *STS*. However, the minimal list, drawn from [OASIS WS-Trust, 2009] is as follows:

Assertion processing error	RECOMMENDED error (Fault code)
The request was invalid or malformed.	`wst:InvalidRequest`
Authentication failed.	`wst:FailedAuthentication`
The specified request failed.	`wst:RequestFailed`
Security token has been revoked.	`wst:InvalidSecurityToken`
Insufficient digest elements.	`wst:AuthenticationBadElements`
The specified RequestSecurityToken is not understood.	`wst:BadRequest`
The request data is out-of-date.	`wst:ExpiredData`
The requested time range is invalid or unsupported.	`wst:InvalidTimeRange`
The request scope is invalid or unsupported.	`wst:InvalidScope`
A renewable security token has expired.	`wst:RenewNeeded`
The requested renewal failed.	`wst:UnableToRenew`

4 WS-FEDERATION

In this SIP, the use of [WS-Federation, 2006] is covered for a web application only, for the issuance of *Security Tokens*, in accordance with [WS-Federation: Passive Requestor Profile, 2003]. The web application is accessed through the use of HTTP 1.1 POST and GET requests, and through the use of HTTP redirection. As can be seen in Figure 1, the client's browser is redirected (using an HTTP 302 status code) to the *STS* (Step 3). This *STS* then authenticates the user (Steps 4 and 5), and redirects the browser back to the original web application with an HTTP POST containing the token for the

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

application (Step 6). This SIP does not mandate the approach for this final step, but it MAY be through the use of JavaScript or some other method.

4.1 Operations

In [WS-Federation: Passive Requestor Profile, 2003], the two REQUIRED operations defined when communicating with the *STS* are:

- Sign in/sign on

  This is the request sent to the web application of the *STS* when using [WS-Federation: Passive Requestor Profile, 2003]. It results in a *Security Token* being returned in the response.

- Sign Out

  This is sent to the web application of the *STS* when the user wishes to sign out. All cookies on the user's machine are then cleared, and cached information about the tokens issued is removed.

  - Other operations from [WS-Federation, 2006], such as Attribute Request and Pseudonym Request are OPTIONAL.
  - The actual uniform resource locators (URL) of these operations are not specified, and MUST be agreed out of band.

4.1.1 Federation Metadata

An *STS* SHOULD publish [WS-Federation, 2006] metadata in order to simplify the configuration and management of identity federation. However, this SIP places no constraints on the distribution of this metadata beyond those contained in the appropriate section (3) of [WS-Federation, 2006], other than to state that the federation metadata MUST be signed.

4.2 Inputs

4.2.1 Sign in

When retrieving a *Security Token* using a browser, [WS-Federation: Passive Requestor Profile, 2003] MUST be used.

The request to the *STS* SHOULD place the parameters in the query string (HTTP 1.1 GET method).

The request to the *STS* MAY place the parameters in form data (HTTP 1.1 POST method).

The *STS* MUST support both GET and POST options.

The following parameters are REQUIRED:

Parameter	Notes
`wa=`*string*	This specifies the action that the requestor is performing. For sign in, this MUST have a value of `wsignin1.0`.
`wtrealm=`*string*	This specifies the target scope for the request.

The following parameters are OPTIONAL for the requestor, but MUST be supported by the *STS*:

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

Parameter	Notes
`wct=`*timestring*	This parameter indicates the current time at the recipient for ensuring freshness, using the XML Schema `datetime` time. The time MUST be in UTC.
`wctx=`*string*	This optional parameter is an opaque context value that MUST be returned with the issued token if it is passed in the request.

4.2.2 Sign out

When submitting a federated sign out request to the STS, the [WS-Federation: Passive Requestor Profile, 2003] profile MUST be followed.

The following parameters are REQUIRED:

Parameter	Notes
`wa=`*string*	This specifies the action that the requestor is performing. For sign out, this MUST have a value of `wsignout1.0`.

The following parameters are OPTIONAL for the requestor, but MUST be supported by the STS:

Parameter	Notes
`wreply=`*URL*	This is the address to which the *STS* MUST redirect the browser if it is present in the request.

4.3 Outputs

The output of the sign-in request to the *STS* is a SAML 2.0 assertion, which MUST have the structure described in [NCIA TR/2012/CPW007253/02, 2012]. The following sections describe the output for the two [WS-Federation, 2006] operations.

4.3.1 Sign in

Security Tokens are returned by passing an HTTP form from the *STS* to the *Relying Party* using the HTTP POST method.

The post to the *Relying Party* SHOULD be initiated through the use of a form submitted to the *Relying Party*.

The form SHOULD be submitted using JavaScript, to hide the process from the user, however it MAY be submitted manually.

If neither of these methods is suitable, then an HTTP 302 Status Code (redirect) MAY be used, though this approach is not supported with all browsers, and so MAY generate a warning message.

As per [WS-Federation: Passive Requestor Profile, 2003], the response MUST be in a form parameter named `wresult`.

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

The contents of the `wresult` parameter must be HTTP-encoded.

The `wresult` parameter MUST contain a `wst:RequestSecurityTokenResponse` element which contains the issued token.

The token MUST be a SAML 2.0 token, as described in [NCIA TR/2012/CPW007253/02, 2012].

As per [WS-Federation: Passive Requestor Profile, 2003], if a context parameter (`wctx`) is passed to the server during the request, then this MUST be returned in the form.

The contents of the `wctx` parameter must be HTTP-encoded.

Once it has received the token, the *Relying Party* MAY use a cookie to maintain trust without having to re-authenticate to the *STS* for every request.

4.3.2 Sign out

When an *STS* receives a `signout` request, it MUST follow the [WS-Federation: Passive Requestor Profile, 2003].

When it has completed the sign out process, the *STS* MUST redirect the browser back to the URL specified in the `wreply` parameter of the request, if present.

If not present, then the *STS* SHOULD redirect back to the original referrer of the sign out request.

4.4 Errors

There are many security-related errors that may be raised as SOAP faults by the *STS*. The actual errors will depend on the implementation of the *STS*. However, the RECOMMENDED list, drawn from [WS-Federation, 2006] is as follows:

Assertion processing error	RECOMMENDED error (Fault code)
No pseudonym found for the specified scope.	`fed:NoPseudonymInScope`
The principal is already signed in (optional – need not be reported).	`fed:AlreadySignedIn`
The principal is not signed in (optional – need not be reported).	`fed:NotSignedIn`
An improper request was made (e.g., Invalid/unauthorized pseudonym request).	`fed:BadRequest`
No match for the specified scope.	`fed:NoMatchInScope`
Credentials provided don't meet the freshness requirements.	`fed:NeedFresherCredentials`
Specific policy applies to the request – the new policy is specified in the S12:Detail element.	`fed:SpecificPolicy`
The specified dialect for claims is not supported.	`fed:UnsupportedClaimsDialect`
A requested RST parameter was not accepted by the STS. The details element contains a fed:Unaccepted element. This element's value is a list of the unaccepted parameters specified as QNames.	`fed:RstParameterNotAccepted`
A desired issuer name is not supported by the STS.	`fed:IssuerNameNotSupported`

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

5 REFERENCES

[IETF RFC 2246, 1999]:
Internet Engineering Task Force Request for Comments 2246, "The TLS Protocol", T. Dierks, C. Allen, IETF, Sterling, Virginia, US, January 1999.

[IETF RFC 4120, 2005]:
Internet Engineering Task Force Request for Comments 4120, "The Kerberos Network Authentication Service (V5)", C. Neuman, T. Yu, K. Raeburn, IETF, Sterling, Virginia, US, July 2005.

[NC3A RD-2814, 2009]:
NATO Consultation, Command and Control Agency Reference Document 2814, "Bi-SC AIS/NNEC SOA Implementation Guidance" (provisional title), J. Busch, S. Brown, R. Fiske, G. Hallingstad, M. Lehman, NC3A, The Hague, Netherlands, unpublished document, December 2009 (NATO Unclassified).

[NCIA TR/2012/CP2007253/02, 2012]:
NATO Communications and Information Agency Technical Report 2012/CP2007253/06, "Security Services Interface Profile Proposal for a Policy Enforcement Point", R. Fiske, M. Lehmann, R. Malewicz, L. Schenkels, D. Gujral, NCIA, The Hague, Netherlands, October 2012 (NATO Unclassified).

[NCIA TR/2012/SPW008000/30, 2012]:
NATO Communications and Information Agency Technical Report 2012/SPW008000/30, "Messaging Service Interface Profile Proposal", R. Fiske, M. Lehmann, NCIA, The Hague, Netherlands, October 2012 (NATO Unclassified).

[OASIS WS-Trust, 2009]:
Organization for the Advancement of Structured Information Standards (on-line), http://www.oasis-open.org, WS-Trust 1.4, at http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.doc, 2 February 2009, viewed 30 March 2011.

[W3C WS-Addressing, 2006]:
World Wide Web Consortium (on-line), http://www.w3.org, Web Services Addressing 1.0 – Core, at http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/, 9 May 2006, viewed 30 March 2011.

[W3C WS-Policy 1.5, 2007]:
World Wide Web Consortium (on-line), http://www.w3.org, Web Services Policy 1.5 – Framework, W3C Recommendation, Asir S. Vedamuthu et al., at http://www.w3.org/TR/2007/REC-ws-poicy-20070904, 4 September 2007.

[WS-Federation, 2006]:
(on-line), Web Services Federation Language (WS Federation), Version 1.1, at http://specs.xmlsoap.org/ws/2006/12/federation/ws-federation.pdf, December 2006, viewed 30 March 2011.

[WS-Federation: Passive Requestor Profile, 2003]:
(on-line), WS-Federation: Passive Requestor Profile Version 1.0, at http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-fedpass/ws-fedpass.pdf, 8 July 2003, viewed 30 March 2011.

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

6 ABBREVIATIONS

CES	Core Enterprise Services
HTTP	Hypertext transfer protocol
IdP	Identity provider
PDP	Policy decision point
SAML	Security assertion markup language
SIP	Service interface profile
STS	Security Token Service
UI	User interface
URI	Uniform resource identifier
URL	Uniform resource locator
UTC	Coordinated universal time
XML	Extensible markup language

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

ANNEX 1 – XML SAMPLES

A.1 MESSAGE SAMPLES (NON-NORMATIVE)

The following represents [OASIS WS-Trust, 2009] messages to and from the *Security Token Service*.

A.1.1 Issue Request Message

```
<soap:Envelope xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
    <wsa:Action soap:mustUnderstand="1">http://docs.oasis-open.org/ws-sx/ws-
trust/200512/RST/Issue</wsa:Action>
    <wsa:MessageID>urn:uuid:c92c65d2-7ccb-4431-b2d2-58871c7e4274</wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
    </wsa:ReplyTo>
  </soap:Header>
  <soap:Body>
    <wst:RequestSecurityToken xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-
trust/200512">
      <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
        <wsa:EndpointReference>
          <wsa:Address>..Target Endpoint..</wsa:Address>
        </wsa:EndpointReference>
      </wsp:AppliesTo>
      <wst14:ActAs xmlns:wst14="http://docs.oasis-open.org/ws-sx/ws-trust/200802">
        <Assertion ID="_717a2735-b0ea-4520-9fc7-18bd334ce2ce" IssueInstant="2011-02-
04T13:20:58.984Z" Version="2.0" xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
          <Issuer>..Issuer Uri..</Issuer>
          <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:CanonicalizationMethod>
              <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#rsa-sha256"></ds:SignatureMethod>
              <ds:Reference URI="#_717a2735-b0ea-4520-9fc7-18bd334ce2ce">
                <ds:Transforms>
                  <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"></ds:Transform>
                  <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"></ds:Transform>
                </ds:Transforms>
                <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"></ds:DigestMethod>

<ds:DigestValue>zplEQyM6b6Za80awbbClZ0A0bHthGshjL98Xe0eD+08=</ds:DigestValue>
              </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>..Signature Value..</ds:SignatureValue>
            <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
              <ds:X509Data>
                <ds:X509Certificate>..Base64 Encoded Issuer
Certificate..</ds:X509Certificate>
              </ds:X509Data>
            </KeyInfo>
          </ds:Signature>
          <Subject>
            <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-
key">
              <SubjectConfirmationData wsa:type="KeyInfoConfirmationDataType"
xmlns:a="http://www.w3.org/2001/XMLSchema-instance">
                <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
                  <xenc:EncryptedKey xmlns:e="http://www.w3.org/2001/04/xmlenc#">
                    <xenc:EncryptionMethod
```

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

```
Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p">
                    <DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
                  </xenc:EncryptionMethod>
                  <KeyInfo>
                    <ds:X509Data xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                      <ds:X509IssuerSerial>
                        <ds:X509IssuerName>..Cert Issuer..</ds:X509IssuerName>
                        <ds:X509SerialNumber>..Cert Ref..</ds:X509SerialNumber>
                      </ds:X509IssuerSerial>
                    </ds:X509Data>
                  </KeyInfo>
                  <xenc:CipherData>
                    <xenc:CipherValue>..Encrypted Key..</xenc:CipherValue>
                  </xenc:CipherData>
                </xenc:EncryptedKey>
              </KeyInfo>
            </SubjectConfirmationData>
          </SubjectConfirmation>
        </Subject>
        <Conditions NotBefore="2011-02-04T13:20:58.977Z" NotOnOrAfter="2011-02-
04T14:20:58.977Z">
          <AudienceRestriction>
            <Audience>..Target Endpoint..</Audience>
          </AudienceRestriction>
        </Conditions>
        <AttributeStatement>
          <Attribute Name="http://schemas.xmlsoap.org/claims/UPN">
            <AttributeValue>..Value from Directory..</AttributeValue>
          </Attribute>
          <Attribute Name="http://ces.nc3a.nato.int/prototype/claims/clearance">
            <AttributeValue>..Value from Directory..</AttributeValue>
          </Attribute>
          <Attribute Name="http://schemas.xmlsoap.org/claims/EmailAddress">
            <AttributeValue>..Value from Directory..</AttributeValue>
          </Attribute>
          <Attribute
Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name">
            <AttributeValue>..Value from Directory..</AttributeValue>
            <AttributeValue>..Value from Directory..</AttributeValue>
          </Attribute>
          <Attribute
Name="http://schemas.xmlsoap.org/ws/2009/09/identity/claims/actor">
            <AttributeValue>
              <Actor>
                <Attribute Name="http://schemas.xmlsoap.org/claims/UPN"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
                  <AttributeValue>..Value from Directory..</AttributeValue>
                </Attribute>
                <Attribute
Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
                  <AttributeValue>..Value from Directory..</AttributeValue>
                  <AttributeValue>..Value from Directory..</AttributeValue>
                </Attribute>
                <Attribute
Name="http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmethod"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion">

<AttributeValue>http://schemas.microsoft.com/ws/2008/06/identity/authenticationmetho
d/windows</AttributeValue>
                </Attribute>
                <Attribute
Name="http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationinstant"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
```

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

```
                    <AttributeValue wsa:type="tn:dateTime"
xmlns:tn="http://www.w3.org/2001/XMLSchema"
xmlns:a="http://www.w3.org/2001/XMLSchema-instance">2011-02-
04T13:20:58.928Z</AttributeValue>
                  </Attribute>
                </Actor>
              </AttributeValue>
            </Attribute>
          </AttributeStatement>
          <AuthnStatement AuthnInstant="2011-02-04T13:18:14.389Z">
            <AuthnContext>

<AuthnContextClassRef>urn:federation:authentication:windows</AuthnContextClassRef>
            </AuthnContext>
          </AuthnStatement>
        </Assertion>
      </wst14:ActAs>
      <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-
trust/200512/Issue</wst:RequestType>
      <wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV2.0</wst:TokenType>
    </wst:RequestSecurityToken>
  </soap:Body>
</soap:Envelope>
```

A.1.1.1 Issue Response Message

```
<soap:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <soap:Header>
    <wsa:Action soap:mustUnderstand="1">http://docs.oasis-open.org/ws-sx/ws-
trust/200512/RSTRC/IssueFinal</wsa:Action>
    <wsa:RelatesTo>urn:uuid:c92c65d2-7ccb-4431-b2d2-58871c7e4274</wsa:RelatesTo>
    <o:Security soap:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsu:Timestamp wsu:Id="_0">
        <wsu:Created>2011-02-04T13:21:02.384Z</wsu:Created>
        <wsu:Expires>2011-02-04T13:26:02.384Z</wsu:Expires>
      </wsu:Timestamp>
    </o:Security>
  </soap:Header>
  <soap:Body>
    <wst:RequestSecurityTokenResponseCollection xmlns:wst="http://docs.oasis-
open.org/ws-sx/ws-trust/200512">
      <wst:RequestSecurityTokenResponse>
        <wst:Lifetime>
          <wsu:Created xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">2011-02-04T13:21:02.357Z</wsu:Created>
          <wsu:Expires xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">2011-02-04T14:21:02.357Z</wsu:Expires>
        </wst:Lifetime>
        <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
          <wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing">
            <wsa:Address>..Target Endpoint..</wsa:Address>
          </wsa:EndpointReference>
        </wsp:AppliesTo>
        <wst:RequestedSecurityToken>
          <EncryptedAssertion xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
            <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
              <xenc:EncryptionMethod
```

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

```
Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc"></xenc:EncryptionMethod>
              <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
                <xenc:EncryptedKey xmlns:e="http://www.w3.org/2001/04/xmlenc#">
                  <xenc:EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p">
                    <DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
                  </xenc:EncryptionMethod>
                  <KeyInfo>
                    <ds:X509Data xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                      <ds:X509IssuerSerial>
                        <ds:X509IssuerName>..Cert Issuer..</ds:X509IssuerName>
                        <ds:X509SerialNumber>..Cert Ref..</ds:X509SerialNumber>
                      </ds:X509IssuerSerial>
                    </ds:X509Data>
                  </KeyInfo>
                  <xenc:CipherData>
                    <xenc:CipherValue>..Encrypted Key..</xenc:CipherValue>
                  </xenc:CipherData>
                </xenc:EncryptedKey>
              </KeyInfo>
              <xenc:CipherData>
                <xenc:CipherValue>..Encrypted SAML Token..</xenc:CipherValue>
              </xenc:CipherData>
            </xenc:EncryptedData>
          </EncryptedAssertion>
        </wst:RequestedSecurityToken>
        <wst:RequestedProofToken>
          <wst:BinarySecret>..Binary Secret..</wst:BinarySecret>
        </wst:RequestedProofToken>
        <wst:RequestedAttachedReference>
          <SecurityTokenReference b:TokenType="http://docs.oasis-
open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0" xmlns="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:b="http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd">
            <KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-
saml-token-profile-1.1#SAMLID">_55696a58-9dd2-46c3-ae65-
37c4b03c32cd</KeyIdentifier>
          </SecurityTokenReference>
        </wst:RequestedAttachedReference>
        <wst:RequestedUnattachedReference>
          <SecurityTokenReference b:TokenType="http://docs.oasis-
open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0" xmlns="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:b="http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd">
            <KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-
saml-token-profile-1.1#SAMLID">_55696a58-9dd2-46c3-ae65-
37c4b03c32cd</KeyIdentifier>
          </SecurityTokenReference>
        </wst:RequestedUnattachedReference>
        <wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV2.0</wst:TokenType>
        <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-
trust/200512/Issue</wst:RequestType>
        <wst:KeyType>http://docs.oasis-open.org/ws-sx/ws-
trust/200512/SymmetricKey</wst:KeyType>
      </wst:RequestSecurityTokenResponse>
    </wst:RequestSecurityTokenResponseCollection>
  </soap:Body>
</soap:Envelope>
```

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

A.1.2 WS-Federation Response Form

```
<html>
  <head>
    <title>..Title..</title>
  </head>
  <body>
    <form method="POST" name="hiddenform"
action="http://fqdn/url_of_relying_party">
      <input type="hidden" name="wa" value="wsignin1.0" />
      <input type="hidden" name="wresult" value="..HTTP
encoded..:<t:RequestSecurityTokenResponse> element.. " />
      <input type="hidden" name="wctx" value="rm=0&amp;id=..HTTP Encoded Context.."
/>
      <noscript>
        <p>Script is disabled. Click Submit to continue.</p>
        <input type="submit" value="Submit" />
      </noscript>
    </form>
    <script language="javascript">window.setTimeout('document.forms[0].submit()',
0);</script>
  </body>
</html>
```