6 Security Composition

6.1 Composition Requirements

To discuss the composition of the facilities provided by WS-ReliableMessaging (WS-RM) with those provided by web service security infrastructures we must first define what we mean by “compose”. For our purposes composition will be defined by the following requirements:

1. The use (or non-use) of WS-RM facilities should not impact a service's ability to provide or require the necessary security qualities (authentication, integrity, confidentiality) in its communications.
2. The use (or non-use) of security services should not impact the service's ability to use WS-RM to provide message tracking and retry.
3. The use (or non-use) of WS-RM facilities should not impose additional security requirements **on individual services**. It may be that, in order to protect the security of the WS-RM protocol itself, additional security infrastructure (e.g. an WS-Trust runtime component) is required, but these additional infrastructure requirements should not impact the services that use WS-RM.

6.2 Profiles

The following sections detail individual security composition profiles. A unique URI is assigned to each profile for use as a reference to that profile.

6.2.1 TLS/BasicAuth Profiles

The following two profiles use TLS/SSL to provide integrity and confidentiality and HTTP Basic Authentication [BasicAuth] to authenticate the RMS. The mechanisms used to protect the Sequence are the same in both profiles, but the scope of the BasicAuth credentials are different. The following applies to both of these profiles:

- These profiles pre-suppose that HTTP is being used to carry all WS-RM messages for the Sequences to which these profiles are being applied.
- The use of TLS limits the application of these profiles to situations in which the source process and the destination can directly connect over a single network hop.

The high-level mechanism of both profiles is as follows:

1. The source process (RMS or combined AS/RMS node) negotiates a server-authenticated TLS session with the destination.
2. The source process authenticates with the destination using HTTP Basic Authentication for all subsequent messages.
3. The RMS establishes a Sequence with the RMD.
4. Protocol and traffic messages for a Sequence may flow over any TLS session between the same source/destination pair.

The specific profiles are:

http://docs.oasis-open.org/ws-rx/wsrmsp/200604/profile/http_auth/samenode

The credentials presented by the source process are meant to serve as authentication information for both the AS and RMS. It is the responsibility of the RMD to forward the source process' authenticated identity to the AD. It may do this by replaying the HTTP Authorization header or by some other, unspecified mechanism.

http://docs.oasis-open.org/ws-rx/wsrmsp/200604/profile/http_auth/sep_node

The credentials presented by the RMS serve solely to authenticate the RMS and do not apply to the AS. The AS may independently authenticate with the AD via some other mechanism, but it cannot use HTTP Basic Authentication to do so as this will conflict with the headers used by the RMS.

Both of these profiles satisfy the security requirements (described in Detailed Security Requirements) as follows:

1-4. The integrity of message bodies and headers and the binding of headers to bodies are all provided by TLS/SSL.

5. At sequence creation time the RMD can perform an authorization check using the identity of the source process (as claimed by the HTTP `Authorization` header that accompanies the `<wsrm:CreateSequence>` request) to determine if the source process is allowed to create Sequences with this RMD.

6-8. The source entity that is permitted to operate on a Sequence is identified by the BasicAuth credentials that accompanied the `<wsrm:CreateSequence>` message for that sequence. Note that it is the entity *referenced* by the credentials and not the actual credentials that is considered in Sequence ownership checks. The destination entity allowed to operate on the Sequence is identified by the X.509 certificate of the authenticated server for the TLS connection used to carry the `<wsrm:CreateSequenceResponse>` message. The use of the TLS session ID to identify the destination entity is discouraged as this binds the lifetime of the Sequence to the lifetime of the TLS session.

6.2.2 TLS/Mutual Authentication Profiles

The following two profiles use TLS/SSL to provide integrity and confidentiality and use client-side certificates to authenticate the RMS. The mechanisms used to protect the Sequence are the same in both profiles, but the scope of the client-side credentials are different. The following applies to both of these profiles:

- These profiles pre-suppose that HTTP is being used to carry all WS-RM messages for the Sequences to which these profiles are being applied.
- The use of TLS/SSL limits the application of these profiles to situations in which the source process and the destination can directly connect over a single network hop.
- The RMD must be configured to trust (either explicitly or through a trust chain) the signer of the source's X.509 certificate.

The high-level mechanism of both profiles is as follows:

1. The source process (AS or combined AS/RMS nodes) negotiates a mutually-authenticated TLS session with the destination.
2. The RMS establishes a Sequence with the RMD over this TLS session.
3. Protocol and traffic messages for a Sequence may flow over any TLS session between the same source/destination pair.
4. Authorization is based on the authenticated identity of the TLS client (as represented by the client-side X.509 certificate). WS-RM protocol and traffic messages for a single Sequence may

flow over any TLS session between the same source/destination pair provided all such sessions share a common client identity. This allows a single Sequence to span multiple TLS sessions.

The specific profiles are:

http://docs.oasis-open.org/ws-rx/wsrmsp/200602/profile/tls_auth/samenode

The identity of the TLS/SSL client represents both the AS and RMS. The RMD is responsible for forwarding the source process' authenticated identity to the AD. The mechanisms by which the RMD accomplishes this are not covered in this document.

http://docs.oasis-open.org/ws-rx/wsrmsp/200602/profile/http_auth/sep_node

The client-side X.509 certificate used to authenticate the RMS applies only to the RMS and should not be construed as representing the identity of the AS. If the AS wishes to authenticate with the AD it must do so using some other mechanism than TLS/SSL mutual authentication.

These profile satisfies the security requirements (described in Detailed Security Requirements) as follows:

1-4. The integrity of message bodies and headers and the binding of headers to bodies are provided by TLS/SSL.

5. At sequence creation time the RMD can perform an authorization check using the identity of the source process (as represented by the client-side X.509 certificate of the TLS/SSL connection over which the `<wsrm:CreateSequence>` request was sent) to determine if the source process is allowed to create Sequences with this RMD.

6-8. The source entity that is permitted to operate on a Sequence is identified by the client-side X.509 certificate of the TLS/SSL connection over which the `<wsrm:CreateSequence>` message for that Sequence was sent. Note that it is the entity *referenced* by the certificate and not the actual certificate that should be considered in Sequence ownership checks. The destination entity allowed to operate on the Sequence is identified by the X.509 certificate of the authenticated server for the TLS/SSL connection used to carry the `<wsrm:CreateSequenceResponse>` message. The use of the TLS/SSL session ID to identify the either the source or destination entities is discouraged as this binds the lifetime of the Sequence to the lifetime of the TLS/SSL session.

6.2.3 WS-SecureConversation Profile

The following profile uses WS-Security [WS-Security] to provide integrity and confidentiality to WS-RM Sequence Lifetime and Traffic Messages. WS-SecureConversation [SecureConversation] is used to mutually authenticate the RMS and RMD. This profile is identified as:

http://docs.oasis-open.org/ws-rx/wsrmsp/200604/profile/wsc_auth

The mechanism for this profile is as follows:

1. The source process uses WS-SecureConversation to create a Security Context between itself and the destination.

2. The source process creates a Sequence with the RMD. The Security Context established in (1) is bound to this via a `<wsse:SecurityTokenReference>` (STR) that refers to the `<wsc:SecurityContextToken>` for that Security Context. This binding is accomplished by including this STR in a `<wsrmsp:SecurityComposition>` extension element (see Profile Agreement and Negotiation below) to the `<wsrm:CreateSequence>` message.

3. All Sequence Lifecycle and Traffic Messages and RM Protocol Header Blocks for the Sequence created in (2) must be signed with the key corresponding to the Security Context created in (1).

All <wsrm:Sequence> headers must be bound to the body of their Traffic Messages by a signature that covers at least the <wsrm:Sequence> header and the message body.

4. Authorization is based on the Security Context communicated in (2). Proof of knowledge of the key(s) underlying the Security Context constitutes permission to operate on any Sequence related to that Security Context. Such proof is provided via WS-Security signatures that are computed using the key(s) in question. For example, if a Sequence Traffic Message has its sequence header bound to its message body by a signature that uses the key associated with the Security Context that is bound to that Sequence, then that message can be considered to have originated from an RMS that is authorized to operate on that Sequence.

Observant readers will note that this profile is not divided into sub-instances to cover the two cases in which the AS and RMS are the same entity and in which the AS and RMS are separate entities. This is due to the fact that WS-Security allows a single mechanism to handle both cases. For example, suppose a source process is composed of a combined AS/RMS entity. The Sequence Traffic Messages issued by this source process contain a single <ds:Signature> element that serves to authenticate both the RMS and the AS. The RMD node can perform all necessary authorization checks against the RMS based on this signature and, if the message is successfully processed, pass the unaltered message to the AD node where it can perform any authorization checks based on the same signature.

Conversely, suppose the AS and RMS are separate, independent entities and that the Sequence Traffic Messages issued by the RMS contain two <ds:Signature> elements, one that was created by the AS and one that was created by the RMS. The RMD node can perform all necessary authorization checks based on the RMS-level signature (it may optionally remove the RMS-level signature from the message). The AD node can perform any authorization checks based on the AS-level signature.

Note that this profile imposes no constraints on and makes no assumptions about the processing of WS-Security defined message properties such as signature or encryption elements. In situations where multiple signatures appear in the same message the determination of which signature corresponds to which source node (RMS or AS) is outside the scope of this profile.

This profile satisfies the security requirements (described in Detailed Security Requirements) as follows:

1-4. The integrity of message bodies and headers are assured and the binding of headers to bodies achieved by including the bodies and headers in WS-Security-defined signatures. These signature are formed using the key(s) corresponding to the Security Context associated with the Sequence under which the Sequence Traffic Message is being carried or to which the Sequence Lifecycle Message (including faults) applies.

5. At sequence creation time the RMD performs an authorization check using the identity of the source process (as conveyed during the establishment of the Security Context used to protect the <wsrm:CreateSequence> message and to which the STR in the <wsrm:CreateSequence> message refers) to determine if the source process is allowed to create Sequences with this RMD.

6-8. The source entity that is permitted to operate on a Sequence is identified by the <wsc:Identifier> of the Security Context that was associated with the Sequence at creation time. The destination entity allowed to operate on the Sequence is identified by the same <wsc:Identifier>. Neither party should refer to specific key instances within a Security Context as this ties the lifetime of the Sequence to the lifetime a specific key.

Ed Note: It's not clear that this profile does enough to address possible interoperability problems associated with using WS-SecureConversation and WS-Security. In particular the number of signatures and the order of their processing seems to be a likely area for conflicts. Perhaps we should constrain the options and simply mandate that the AS and RMS represent themselves as separate entities (i.e. require two signatures) even if they share the same process/address-space? We could also require adherence to Basic Security Profile.