SAML credentials-collector use-cases and requirements

Working draft 01, 30 September 2003

Document identifier: oasis-sstc-v2.0-credentials-collector-use-cases-wd-01

Location: http://www.oasis-open.org/committees/documents.php?wg_abbrev=security

Send comments to: security-services-comment@lists.oasis-open.org

Editors:

Tim Moses, Entrust

Contributors:

Jeff Hodges, Sun Microsystems
Hal Lockhart, BEA

Abstract:

This working draft defines use-cases and requirements for a protocol by which system entities can authenticate to a credentials-collector with the help of an authentication authority.

Status:

This version of the specification is a working draft of the committee. As such, it is expected to change prior to adoption as an OASIS standard.

If you are on the security-services@lists.oasis-open.org list for committee members, send comments there. If you are not on that list, subscribe to the security-services-comment@lists.oasis-open.org list and send comments there. To subscribe, send an email message to security-services-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

Table of contents

1. Introduction

This document describes use-cases and requirements for a protocol by which a system entity can authenticate bi-laterally with a credentials-collector, using authentication services provided by an authentication authority.

The protocol is intended to support a broad range of authentication mechanisms, including those that result in a secret shared between the system entity and the credentials collector for protecting a subsequent session.

2. Use-cases

2.1. Use-case 1: Authn authority as credentials-collector

Use-case 1 is shown in Figure 1.

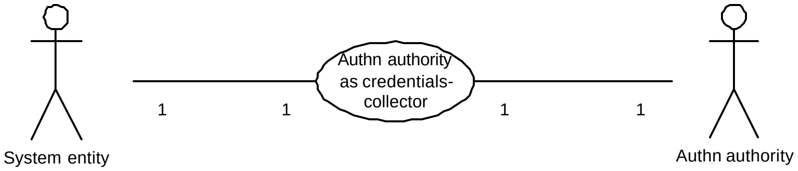


Figure 1 - Use-case 1

In this use-case the functions of the credentials-collector are performed by the authentication authority.

The state transition diagram is shown in Figure 2.

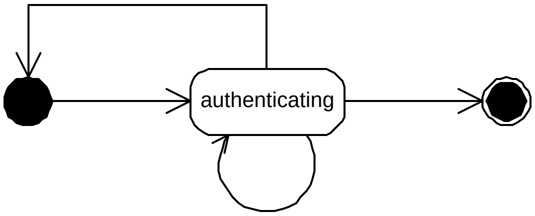


Figure 2 - State transition diagram

In this diagram, the initial state corresponds to the unauthenticated state and the final state corresponds to the authenticated state.

The sequence of activities in the “authenticating” state is shown in Figure 3.

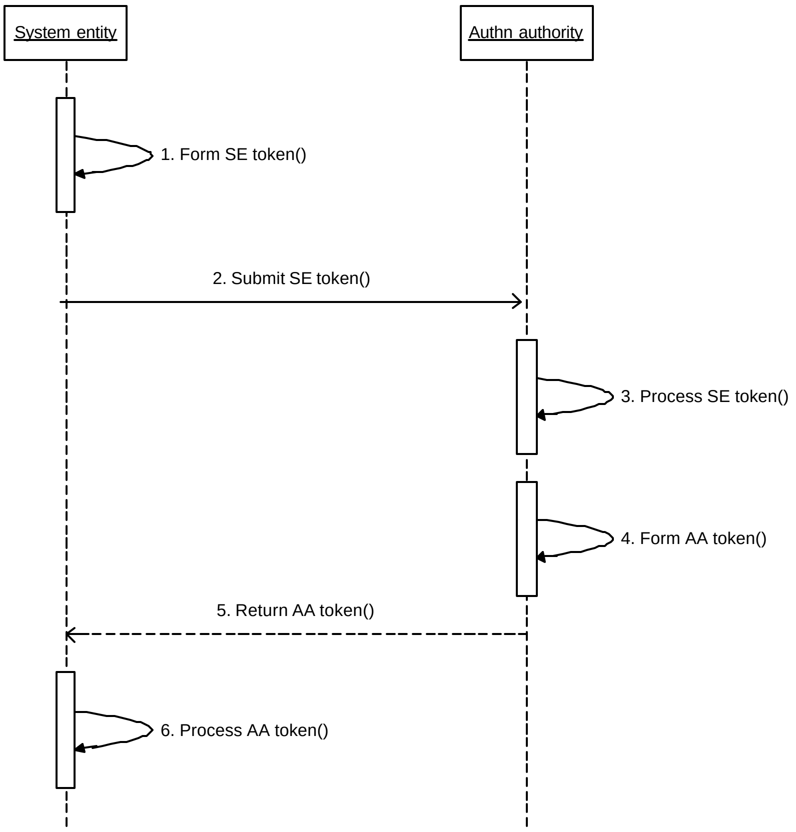


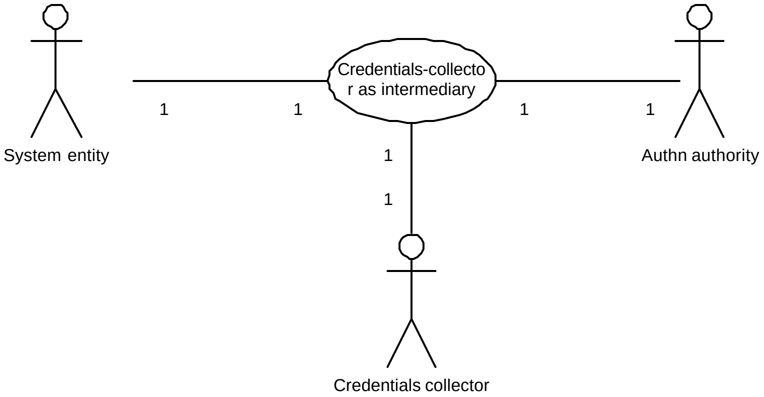
Figure 3 - Use-case 1 “authenticating” sequence

1. System entity forms an authentication token (this step is optional on the first occurrence).
2. System entity sends the token to authentication authority.
3. Authentication authority processes the token and decides whether system entity is authenticated or not.
4. If system entity is not authenticated, then authentication authority forms a token. Optionally, if system entity is authenicated, then the token is an authentication assertion.
5. Authentication authority returns the token to systeme entity.
6. System entity processes the token.

[SASLib] specifies a solution for this use-case, based on the Simple Authentication and Security Layer.

2.2. Use-case 1: Credentials-collector as intermediary

Use-case 2 is shown in Figure 4.



71

72 **Figure 4 - Use-case 2**

The state transition diagram is identical to the one shown in Figure 2.

The sequence of activities in the “authenticating” state is shown in Figure 5.

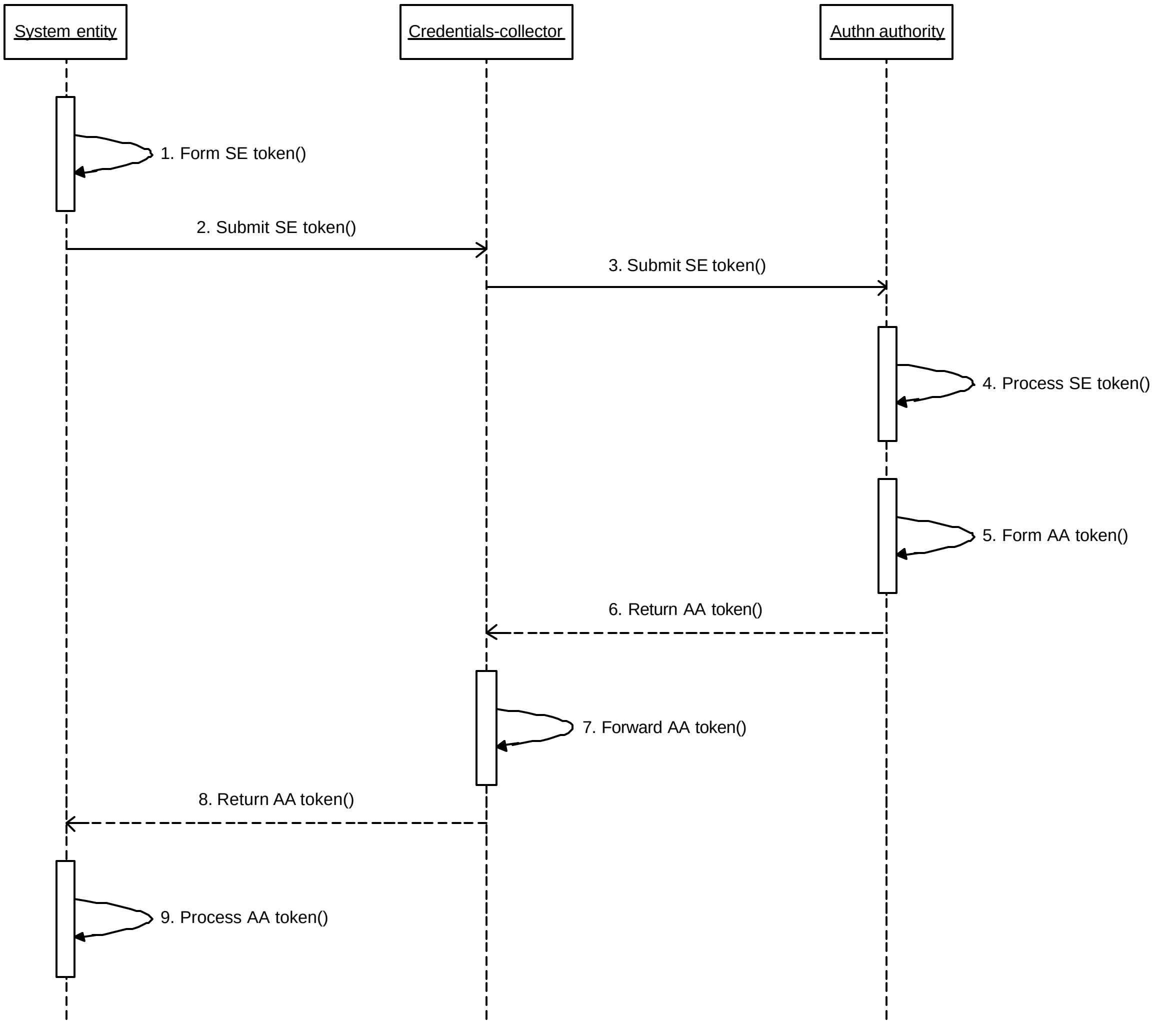


Figure 5 - Use-case 2 sequence

1. System entity forms an authentication token (this step is optional on the first occurrence).
2. System entity sends the token to credentials-collector.
3. Credentials-collector forwards the token to authentication authority.
4. Authentication authority processes the token and decides whether system entity is authenticated or not.
5. If system entity is not authenticated, then authentication authority forms a token. If system entity is authenicated, then the token is an authentication assertion.
6. Authentication authority returns the token.

7. If the authentication is complete, then credentials-collector extracts the authentication assertion.

8. If authentication is not complete, then credentials-collector forwards the token to system entity.

9. System entity processes the token.

In the case where the authentication mechanism results in a secret shared between the authentication authority and the system entity, the resulting secret is passed to the credentials-collector in the subject-confirmation element of the authentication assertion.

[SASLib] does not currently address this use case. However, with straightforward modification, it could be made to address it.

10. Requirements

In the case where the shared secret is a secret-key, its confidentiality must be protected in step 6 of Section 2.2. This may be accomplished by means of a secure session between the authentication authority and the credentials-collector or by encrypting the shared-secret for the credentials-collector. The former case may not be suitable if the credentials-collector forwards the assertion. In this case, the credentials-collector could remove the shared secret from the assertion. But, this may disrupt any integrity protection applied to the assertion by the authentication authority.

For these reasons, encrypting the shared-secret for the credentials-collector may be the most appropriate solution.

11. References

[SASLib] Liberty SASL-based SOAP Authentication Specification, Version: 1.0-03, Jeff Hodges, located at: http://www.projectliberty.org/specs/draft-lib-arch-soap-authn-v1.0-03.pdf

Appendix A. Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS has been notified of intellectual property rights claimed in regard to some or all of the contents of this specification. For more information consult the online list of claimed rights.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

This document and the information contained herein is provided on an “AS IS” basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.