Kerberos & SAML Interoperability Framework

Oasis SSTC

DRAFT v.0005

Thomas Hardjono & Josh Howlett

Table of Contents

Kerberos and SAML Interoperability Framework

1. Overview

The current document seeks to address interoperability requirements between the Kerberos v5 authentication protocol and SAML2.0 systems within the context of web security. The overarching goal of the work is to bring the Kerberos authentication protocol to the web, allowing the Kerberos protocol and its components to be more readily integrated into SAML2.0 systems.

The approach taken in this document is to identify some key use-cases, identify some technical requirements for interoperability, and present solutions in the form of SAML2.0 profiles and other constructs that allow a SAML2.0 system to make use of Kerberos.

The current document also seeks to be an open discussion platform for the SAML2.0 community within the Oasis SSTC and other Oasis WGs, and thus it seeks technical input and guidance from these communities.

2. Definitions, Acronyms and Abbreviations

Abbreviations	Definition
SAML	
SAML Attribute Authority	
SAML Requester	
Kerberos AS	
Kerberos TGS	
Service Principal	
Client Principal	
TBD	

NB: Copy some entries from SAML Glossary doc as main glossary source.

3. References

[1] J. Hodges, J. Howlett, L. Johansson & R.L. Bob Morgan, *Towards Kerberizing Web Identity and Services*, MIT Kerberos Consortium, January 2009.

[2] R. Needham & M. Schroeder, "Using encryption for authentication in large networks of computers.", *Communications of the ACM* 21 (12): 993-999, December 1978.

[3] Jennifer G. Steiner, Clifford Neuman, Jeffrey I. Schiller, “Kerberos: An Authentication Service for Open Network Systems”, *Usenix Conference Proceedings*, 1988, pp.191-200.

[4] SAML2.0 CORE doc.

[5] SAML2.0 Glossary doc.

[6] SAML2.0 Public Key HoK doc.

[7] Liberty Alliance, *Liberty ID-FF Architecture Overview* (v1.2).

[8] Liberty Alliance, *Liberty ID-WSF Web Services Framework* (v2.0).

[9] WS-Security Kerberos Token Profile 1.1

[10] Microsoft KILE extensions doc [MS-KILE]

[11] Microsoft S4U extensions doc [MS-SFU]

4. Background: Kerberos Authentication Protocol

4.1. The Kerberos Authentication Paradigm

The Kerberos authentication protocol was one of the earliest authentication protocols to recognize the need for (and incorporate) the notion of attributes of the end-user as part of authorization decision-making. Thus although the fundamental Needham-Schroder protocol 3 underlying Kerberos did not include the notion of tickets or tokens, the use of Kerberos within Project Athena 3 resulted in the introduction of tickets. The basic interaction is summarized within Figure 1 in which a *Requestor* entity (Kerberos client) seeks access to a given resource belonging to a *Relying Party* entity (Service). The Requestor is required to authenticate itself to a *Verifier* entity, who in-turn issues a Ticket as proof of successful identification and authentication of the Requestor entity. The Requestor then presents the Ticket to the Relying Party in order to obtain access to resource or services. An additional Ticket Granting Ticket (TGT) layer can be added upon this model to provide further convenience to the Requestor.

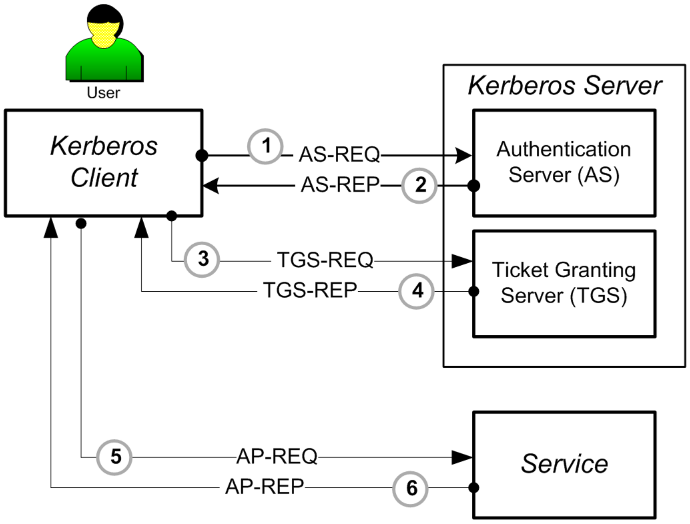

Figure 2-1: Basic Kerberos Authentication Protocol

MORE TEXT (Thomas)

4.2. The Kerberos Authentication Protocol

MORE TEXT (Thomas)

5. Use-Case: Web Single-Sign-On

Web Single-Sign-On (Web-SSO) today provides a key building block for building secure web-based systems. Currently the main authentication method in Web-SSO is the password-over-forms, which are typically transmitted over a TLS connection (which in the majority of the case operates without user X.509 certificates, either self-signed or CA-issued).

In this section we look at Web-SSO as defined in the Oasis SAML2.0 specifications, which became an Oasis standard in 2005. Derived from a combination of SAML1.1,the ID-FF Specifications from the Liberty Alliance (Project Liberty), and the Shibboleth Architecture from the Internet2 community, the SAML2.0 specifications include profiles and binding that can be used to address Web-SSO.

5.1. Web-SSO Architecture

The SAML2.0 Web-SSO model and flows are shown in Figure 3-1. Here, in Steps (1) and (2) the Client (e.g. Browser), which seeks access to a resource at the Service Provider (SP), is redirected by the SP to the Identity Provider (IdP).

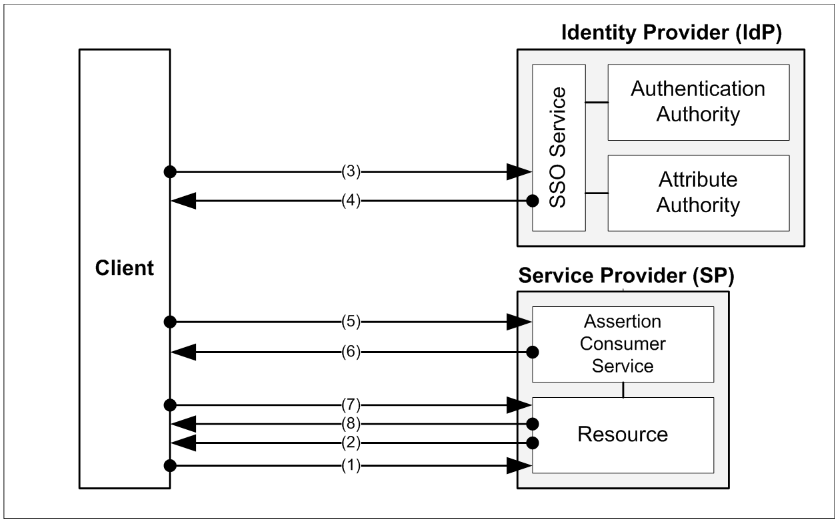

Figure 3-2: Summary of the SAML2.0 Web-SSO Architecture

The Client then requests the SSO Service from the IdP in Step (3), and the SSO Service attempts to authenticate the client (for example, by returning an HTML form to the Client and authenticating the credentials provided in the submission of it; or authenticating a user certificate). The IdP returns a SAML authentication assertion to the Assertion Consumer Service at the SP in Step (5). Upon successful verification of the assertion, the Client is redirected to the original resource again (Step (7)) and the SP responds with the requested resource in Step (8).

5.2. Role of Kerberos in Web-SSO

In mapping the traditional Kerberos entities to the SAML2.0 Web-SSO entities, its is possible – but not necessarily practical – to map the Kerberos KDC to the IdP. This is largely because the IdP is itself a service in the sense of a "service principal" in Kerberos.

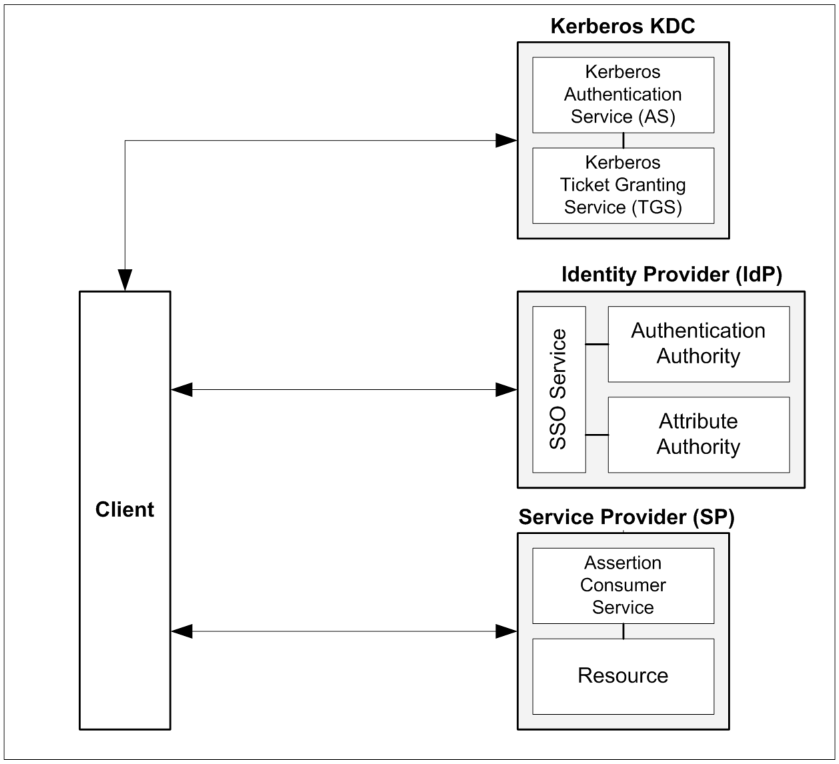

Figure 3-3: KDC in the Web-SSO Architecture

Thus, a better mapping is one in which the IdP expects to see Kerberos Service-Tickets which have been issued by the Ticket Granting Service (TGS) within the KDC (Figure 3-3). There are a number of motivations for this approach:

- *Expanded use of Kerberos Service-Tickets*: Many organizations such as Enterprises wish to maintain their current KDC infrastructure deployment but expand the usage of service-tickets beyond the classic Kerberos usage models. Thus, the thinking here is that once a user (e.g. employee) has succeeded in obtaining a service-ticket from the KDC (i.e. the TGS inside the KDC), the user should be able to present this service-ticket to various services,including web-based services, without being prompted to authenticate again.
- *Decoupling authentication from authorization*: The approach of viewing the IdP as another Service Principal (to Kerberos) opens-up the possibility for the IdP Attribute Authority to embellish a SAML assertion with Kerberos semantics, which may be useful evidence for authorization and access control. In a sense, the IdP Attribute Authority becomes a "Kerberized service" that manages information (regarding Kerberos Clients) which is crucial for authorization and access control down-stream at the various SPs. This decoupling of the KDC as an authentication entity from any authorization function allows the Kerberos KDC to fit into and function in a much broader range of architectures within a broader set of use-cases.

- *Lower barrier to deployment*: The approach of viewing the IdP as another Service Principal (to Kerberos) allows organizations to add additional SAML2.0 entities into the infrastructure without affecting their existing Kerberos infrastructure, which in many organizations is the first port-of-call for employees in their daily work. This approach requires that the IdP Authentication Authority and Attribute Authority to process Kerberos service-tickets and map these into the relevant SAML2.0 assertions. Standardizing this missing component is an important step in bringing Kerberos to the Web, and is addressed further below (See Section 5.3).

5.3. Bridging the Gap: Kerberos Holder of Key

An important role that the IdP has within the SAML2.0 world is that of a SAML issuer. In the Kerberos-Web case it is desirable that the IdP (Authentication Authority) have the ability to issue SAML assertion which are bound to attesting entities by incorporating Kerberos-based evidence that allows a consumer of the assertion to confirm that the attesting entity is authorized to perform this role. This can raise the level of trust that a consumer of an attribute can give to an assertion because, for example, it prevents Man-In-The-Middle attacks (such as a replay attack by an unauthorized attesting entity). We refer to this as a *Kerberos Holder-of-Key assertion* (HoK). Currently, there is already a SAML2.0 HoK Profile for public keys (REF[]). An equivalent HoK profile is required for Kerberos service keys (ie. a symmetric key).

In discussing the HoK Assertions, it is important to note that the IdP generates the HoK Assertions independent from the mechanisms that the Client uses to authenticate against the IdP. That is, the authentication instance and mechanism that the Client happens to use to engage with the IdP is independent from the HoK Assertions pertaining to the service-ticket possessed by the Client. This is because the IdP is making assertions only about the Client Principal possessing the Kerberos Service-Key (of which an encrypted copy resides in the service-ticket). Thus, in effect the Client could be using plain username/password within an authentication event/instance with the IdP, while at a later time it could be using Kerberos or another form of authentication (e.g. X.509 certs or OTP).

This non-dependence of the IdP (on using Kerberos itself to authenticate a Client) provides the greatest flexibility for the HoK Assertions to be used by various services offered by SPs. The role of the IdP is thus to bind the Kerberos principal name of the authenticated identity (using whichever authentication mechanism is desired, which may or may not be Kerberos) to a `<saml:SubjectConfirmation>` element within an holder-of-key assertion.

The HoK Assertion is used by the Client Principal when it seeks access to resources or services to a Relying Party (e.g. a Service Provider). In this case, the Client authenticates to the RP using the Kerberos protocol. In addition to presenting the service-ticket to the RP as part of the usual Kerberos protocol, the Client must also present the HoK Assertion. Here, the RP has the important task of verifying that the two identifies match. That is, the RP must ensure that the Client Principal identifier found in the service-ticket matches the Client identifier found in the HoK Assertion.

The HoK Assertion is in effect stating that the Client Principal is a known entity to the KDC (in its realm) through the fact that the KDC is sharing a symmetric-key with the Client Principal.

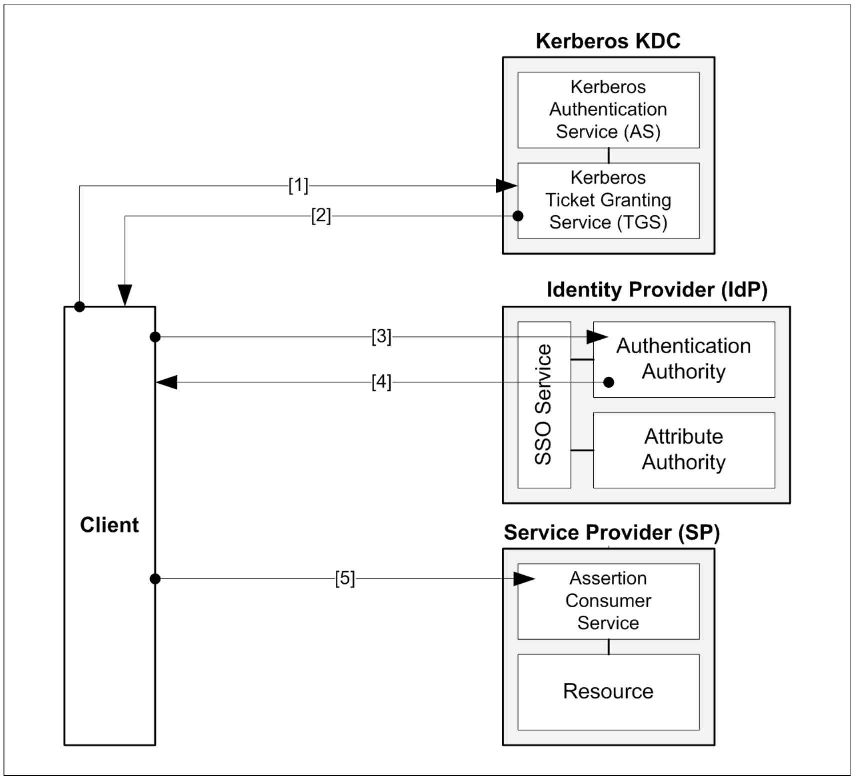

Figure 3-4: Kerberos Holder-of-Key (HoK) Assertion usage model

Figure 3-4 summarizes the usage model of a Kerberos HoK assertion:

- After obtaining the usual Kerberos service-ticket from the KDC (in Steps (1) and (2)), the Client delivers the Kerberos service-ticket to the IdP is Step (3). Note that there a number of steps omitted related to the Client's attempt to obtain access to resources at the SP (as shown in the previous Figure 3-2).

- It is important to point out that in Step (3) the IdP need not necessarily use the Kerberos protocol to authenticate the Client. Here, the service-ticket contains among others the Client Principal name and the service-key (encrypted using the IdP long term Kerberos key). Note that if Kerberos is not used, then the IdP must use some kind of mapping or look-up in order to map between the authenticated identity (say, a certificate CN) and the Kerberos principal.

- The IdP Authentication Authority decrypts the relevant portions of the Client's Service-Ticket, and upon successful verification concludes that the Client is the true principal to which the KDC issued the service-ticket and that the Client is in possession of the relevant Kerberos service-key.

- In Step (4) the IdP Authentication Authority issues a HoK Assertion, naming the Client Principal within the `<saml:SubjectConfirmation>` field of the HoK assertion.

- In Step (5) when the Client authenticates to a Relying Party (e.g. the SP) using the Kerberos protocol, in addition to presenting the service-ticket the Client must also present the HoK Assertion.

There are a number of advantages to such a *Holder-of-Key* (HoK) assertions:

- TBD.

6. Use-Case: Web-Service access of Kerberized Services

In this section we address the use-case pertaining to access to a *Kerberized Service* via a web-service. In general terms, we want to address the use-case in which a SAML system entity requires access to a local/remote Kerberized Service on a behalf of a Client (user) Principal. The term "Kerberized Service" denotes a Service Principal in the Kerberos terminology, and which by definition requires a Kerberos service-ticket to access.

6.1. Accessing Kerberized Services

In order to explain better this use-case, Figure 4-1 shown an example of a human user that wishes to access his/her email remotely using a Web-Mail Service. The user's actual mail-server is an IMAP Server that is Kerberized. The Web-Mail is a service that is operating on behalf of the user/client (not shown in the figure).

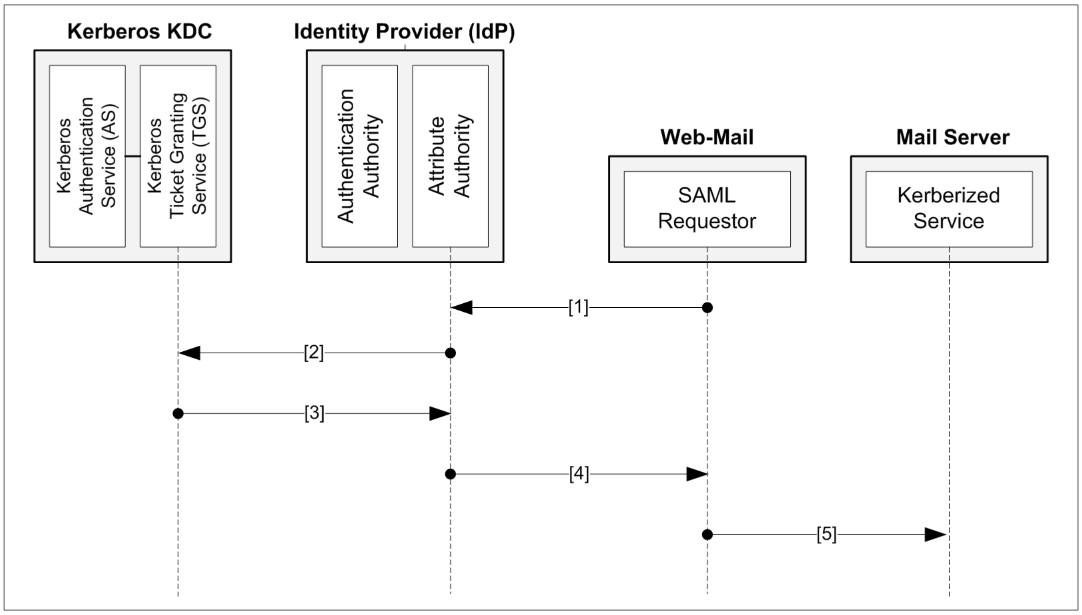

Figure 4-5: Example of Web-Services accessing a Kerberized Service

One of the major issues in Figure 4-1 is the fact that the SAML Requestor may not be able to request a service ticket directly from the KDC since it is an entity that is not recognized by the KDC. In the following section we propose the use of the SAML2.0 Assertion Query Protocol and Request Protocol to address this use-case, together with the S4U2proxy extension [MS-SFU].

6.2. The S4U Extension in Kerberos

The *Service-for-User* (S4U) is a Microsoft extension to the Kerberos protocol, which is also supported by the MIT Kerberos code-base starting in its Release v1.7.

Service for User (S4U) specifies two extensions to the Kerberos Protocol [MS-SFU]. Together the two extensions allow an application service to obtain a Kerberos service ticket on behalf of a user.

The service-ticket produced from an S4U exchange can be used for [MS-SFU]:

- The requesting service's own information.
- Access control local to the service's machine (through impersonating the user).
- Requests to some other service through impersonating the user.

The S4U extension is actually composed of two (2) extensions [MS-SFU]:

a. *S4U2self*: The S4U2self (Service-for-User-to-Self) extension allows a service to obtain a Kerberos service-ticket to itself on behalf of a user. This enables the service to obtain the user's authorization data that is then used in authorization decisions in the local service.

b. *S4U2proxy*: The S4U2proxy (Service-for-User-to-Proxy) extension enables a service to obtain a service ticket on behalf of the user to a second, back end service. This allows back-end services to use Kerberos user credentials as if the user had obtained the service ticket and sent it to the back end service directly. Local policy at the ticket-granting service (TGS) can be used to limit the scope of the S4U2proxy extension.

6.3. Using the SAML2.0 Assertion Query and Request Protocols

The approach to solving this use-case based on the S4U Extension to Kerberos is summarized as follows (see Figure 4-5):

[Step-1] Not having a service-ticket to access the user’s mail-server, the Web-Mail application (denoted as the SAML Requestor) sends an `<AttributeQuery>` to the SAML Attribute Authority (within the IdP), identifying the Kerberos Client Principal (ie. the user) and the target Kerberized Service (namely the user’s Kerberized mail-server).

[Step-2] The IdP Attribute Authority processes the query, and in-turn sends a S4U2proxy request to the TGS within the KDC. This is in essence a request for a Kerberos service-ticket to itself (i.e. to the IdP) on behalf of the Client Principal (i.e. user).

[Step-3] Upon successful validation of the S4U2proxy request from the IdP, the TGS issues a service-ticket, setting the *forwardable* flag within the ticket and placing the Client Principal’s name (instead of the IdP) within the service-ticket. Later this would allow the service-ticket to be accepted by the Kerberized Service.

[Step-4] Upon receiving the service-ticket from the TGS, the IdP delivers this service-ticket to the SAML Requestor (i.e. the Web-Mail application).

[Step-5] The SAML Requestor presents the service-ticket to the Kerberized Service (i.e. Kerberized mail-server) as part of the access request.