Federación de servicios kerberizados en eduroam

Alejandro Pérez Méndez[1], Fernando Pereníguez García[1], Rafael Marín López[1], Gabriel López Millán[1]
[1]Dept. Ingeniería de la Información y las Comunicaciones (DIIC)
Facultad de Informática, Universidad de Murcia, 30100, Espinardo, Murcia
Tel.: +34 868 884 644 Fax: +34 868 884 151
Email: {alex,pereniguez,rafa,gabilm}@um.es

***Abstract*—Eduroam se ha convertido en uno de los principales ejemplos mundiales de federación de red, donde cientos de instituciones permiten que los usuarios en *roaming* pertenecientes a otras instituciones miembro accedan al servicio de red. En este contexto, este artículo describe cómo un usuario de eduroam puede acceder a servicios federados haciendo uso del protocolo Kerberos, aprovechando la autenticación realizada durante el acceso a la red. De esta forma se evita que el usuario tenga que autenticarse para cada uno de estos servicios. Además, ofrece la posibilidad de realizar autorización avanzada para proporcionar servicios de valor añadido (ej. QoS).**

I. Introducción

Las federaciones de identidad han sido uno de los tópicos de seguridad que más importancia ha ganado en la última década. Desde la aparición de tecnologías como SAML [1], OpenID [2] y OAuth [3], se han desplegado gran cantidad de soluciones para permitir que los usuarios accedan a los servicios de Internet haciendo uso de identidades federadas. Es decir, un único proceso de autenticación permite el acceso a múltiples servicios. Algunos ejemplos de comunidades federadas son Google Apps [1], Yahoo [2] o InCommon [3].

Aunque la mayoría de estas soluciones están centradas en los servicios web, existen algunos casos donde la federación de identidades se ha desplegado para el acceso a la red. El ejemplo más significativo es eduroam [4], presente en cientos de instituciones de todo el mundo y centrada en la comunidad investigadora internacional, que permite que usuarios en *roaming* accedan al servicio de red desde cualquier otra institución que pertenezca a la federación usando las credenciales definidas en su organización origen.

Eduroam está basada en una infraestructura AAA (*Authentication, Authorization, Accounting*), basada en RADIUS [5], donde el usuario se autentica para el acceso a la red en la institución visitada mediante el uso de las tecnologías 802.11 [6] y EAP [7].

Como resultado del éxito de eduroam, han surgido diversas iniciativas que tratan de proporcionar servicios de seguridad de valor añadido sobre esta federación. Una de estas iniciativas es DAMe [8] (*Deploying Authorization Mechanisms for federated services in eduroam architecture*), desarrollada bajo el ámbito de los proyectos GEANT [9]. El propósito de DAMe es proporcionar servicios de autorización avanzada en eduroam, teniendo en cuenta atributos de usuario obtenidos de su institución origen (ej. rol, puesto, etc.), que se evalúan para determinar si se concede o deniega el acceso a la red. Además, se distribuye un token de autenticación (*eduToken*) al usuario, que puede usarse para solicitar acceso a otros servicios basados en web, consiguiendo de esta forma una solución SSO (*Single Sign-On*) inter-capa o *cross-layer SSO*.

No obstante, sería deseable extender esta federación de identidad y proceso de SSO a cualquier servicio de Internet desplegado por las instituciones eduroam, tales como SMTP, SSH, XMPP, etc. De esta forma, una vez que el usuario ha sido autenticado para acceder a la red, podría acceder a esos servicios sin necesidad de realizar otro proceso de autenticación. Además, se podría mejorar sensiblemente la experiencia de usuario, permitiendo a las instituciones ofrecer servicios diferenciados basados en los atributos de usuario y no únicamente en su nombre y contraseña. Por ejemplo, los investigadores en *roaming* podrían tener acceso a servicios restringidos, dependiendo de su rol en sus instituciones origen, o servicios diferenciados para alumnos visitantes.

Este trabajo propone una solución a este problema, teniendo en cuenta que la mayoría de los servicios existentes soportan el uso de Kerberos [10]. Se puede utilizar este protocolo para controlar el acceso a los servicios, aprovechando el eduToken distribuido durante el acceso a la red y el material criptográfico resultante del proceso EAP para pre-autenticar al usuario con el KDC (Kerberos Distribution Center). De esta forma se extiende el soporte *cross-layer SSO* a cualquier servicio que soporte Kerberos. Además, el proceso de autorización se realiza de una forma transparente para los servicios, ya que es llevado a cabo por el KDC.

El resto de este artículo se estructura de la siguiente manera. La sección II proporciona un breve resumen de las principales tecnologías relacionadas, mientras que la sección III describe el trabajo relacionado. La sección IV describe la arquitectura propuesta, y la sección V detalla la operación general de la arquitectura. Finalmente, la sección VI presenta algunas conclusiones y vías futuras.

II. Estado del arte

A. Kerberos

Kerberos [10] es un protocolo de seguridad para autenticación y distribución de claves basado en criptografía simétrica que proporciona un acceso SSO a servicios de aplicación mediante el uso de *tickets*. Kerberos define un

[1]http://www.google.com/apps/intl/en/group/index.html
[2]http://www.yahoo.com
[3]http://www.incommonfederation.org

modelo de control de acceso donde un *cliente* que desea acceder a un *servicio* (denominado *servidor de aplicaciones*) debe acudir a un *Key Distribution Center* (KDC) encargado de autenticar y proporcionar tickets a usuarios. Kerberos asume la existencia de ciertas relaciones de confianza entre estas tres entidades. En concreto, el KDC comparte una clave secreta con el cliente y el servicio, respectivamente. En particular, la clave secreta compartida entre cliente y KDC (denominada *reply key*) es derivada típicamente de un password.

Kerberos define tres tipos de intercambio. Inicialmente, por medio del intercambio KRB_AS_REQ/REP, el usuario solicita al KDC ser autenticado para obtener un *Ticket Granting Ticket* (TGT). En este intercambio, Kerberos ofrece un framework extensible de *pre-autenticación* [11] que permite la definición de nuevos mecanismos de autenticación de usuario. A continuación, cuando el cliente desea acceder a un servicio, debe realizar dos tipos de intercambio. En primer lugar, empleando el TGT, contacta con el KDC a través de un intercambio KRB_TGS_REQ/REP para obtener un *Service Ticket* (ST). Finalmente, mediante el intercambio KRB_AP_REQ/REP, el cliente contacta con el servicio y solicita acceso presentando el ST previamente obtenido. Un análisis detallado de las propiedades que hacen de Kerberos un protocolo seguro puede encontrarse en [12].

Kerberos soporta otro modo de operación denominado *cross-realm*, que permite a un cliente acceder a un servicio no controlado por el KDC donde el usuario está registrado. Sin embargo, su uso no está extendido ya que requiere establecer relaciones de confianza directas entre KDCs de distintos dominios, incluso de dominios intermedios no interesados en proveer servicios *kerberizados*.

B. EAP: Extensible Authentication Protocol

El protocolo EAP [7] proporciona un *framework* de autenticación flexible para el acceso a la red. Esta flexibilidad se consigue por medio a los llamados *métodos EAP* que permiten autenticación de usuario utilizando diversos mecanismos como claves simétricas o certificados.

Una autenticación EAP consiste en la ejecución de un método EAP concreto entre el *EAP peer* y el *EAP server* a través del *EAP authenticator*. Mientras que el *EAP peer* es implementado en el dispositivo del usuario, el *EAP server* se ubica en un servidor de AAA. El *EAP authenticator* se localiza en un dispositivo de acceso a la red como puede ser un punto de acceso. El transporte de los paquetes EAP se realiza a través de un protocolo *EAP lower-layer* (ej., 802.11) entre el *EAP peer* y al *EAP authenticator*, mientras que esta funcionalidad es acometida por un protocolo AAA (ej., RADIUS [5]) entre el *EAP authenticator* y el *EAP server*.

La autenticación EAP es iniciada por el *EAP authenticator* cuando solicita la identidad del usuario mediante un mensaje *EAP Request/Id*, quien contesta con un *EAP Response/Id*. A continuación tiene lugar la ejecución del método específico EAP (ej., EAP-TLS) que consiste en varios intercambios *EAP Request/Response* entre *EAP peer* y *EAP server*. La autenticación finaliza cuando el *EAP server* genera un *EAP success* que es enviado el *EAP peer*.

La mayoría de métodos EAP generan material criptográfico compuesto por dos claves: *Master Session Key* (MSK) y *Extended Master Session Key* (EMSK) [13]. Mientras que la MSK se usa para establecer una asociación de seguridad entre el EAP peer y authenticator, la EMSK se reserva para generar una jerarquía de claves entre EAP peer y server siguiendo el proceso descrito en [14].

C. eduroam/DAMe

DAMe [8] es una extensión a la infraestructura *eduroam* que soporta autorización de usuario tanto durante el acceso a la red como a servicios de aplicación. Las extensiones de DAMe se basan en los estándares SAML [1] y XACML [15] para representar sentencias de autenticación/atributos e información de políticas, respectivamente. Adicionalmente, DAMe propone un *framework cross-layer SSO* con el objetivo de conseguir un acceso a servicios partiendo de una única autenticación realizada durante el acceso a la red.

La arquitectura *eduroam/DAMe* opera de la siguiente manera. En primer lugar, un usuario ejecuta un control de acceso a la red basado en EAP que involucra a los servidores RADIUS de las instituciones visitada y origen. Si el usuario es autenticado correctamente, el *Identity Provider* (IdP) de la institución origen genera un *statement* SAML de autenticación (*eduToken*) que es proporcionado al usuario. El eduToken tiene asociado un pseudónimo para identificar el usuario en futuras comunicaciones con el IdP. En particular, durante el acceso a la red, este pseudónimo es empleado por el servidor RADIUS de la institución visitada para solicitar al IdP atributos del usuario, que son proporcionados al *Policy Decision Point* (PDP) local para tomar una decisión de autorización. El eduToken es almacenado por el usuario y usado posteriormente para conseguir un acceso SSO a servicios web de la federación. Una descripción más detallada puede encontrarse en [16] .

III. Trabajo relacionado

Alcanzar un acceso federado a servicios es un aspecto atractivo para la comunidad investigadora. Por ejemplo, los autores en [17] proponen un modelo de control de acceso federado a servicios basado en Kerberos y EAP, asumiendo la existencia de una infraestructura AAA. En lugar de seguir el modelo tradicional de Kerberos donde se exige que el usuario esté registrado de antemano con el KDC para obtener el TGT, se define un mecanismo de pre-autenticación Kerberos donde el KDC verifica la identidad del usuario mediante una autenticación EAP. Sin embargo, esta aproximación no ofrece una solución cross-layer SSO debido a que asume dos autenticaciones EAP independientes: una para obtener acceso a la red, y otra para ser autenticado por el KDC y obtener un TGT.

Por otra parte, el acceso federado a servicios de aplicación también está siendo abordado por el proyecto Moonshot [18], desarrollado bajo el ámbito de TERENA

EMC2[4] y asistido por un nuevo grupo de trabajo dentro del IETF (*Internet Engineering Task Force*) llamado ABFAB (*Application Bridging for Federated Access Beyond Web*). Asumiendo que una gran mayoría de los actuales servicios de aplicación (ej., HTTP, SSH, etc.) soportan GSS-API (RFC 2743) como servicio de autenticación, Moonshot está trabajando en la definición de un nuevo mecanismo GSS-API basado en EAP. De este modo, EAP es empleado por la aplicación para autenticar al usuario, donde los mensajes EAP son transportados dentro de tokens GSS-API. La aplicación contactará el servidor AAA origen (por medio de la infraestructura AAA) para solicitar la validación de los mismos. Sin embargo, este modelo de federación no proporciona un acceso SSO que permita unificar la autenticación de acceso a la red y servicio. De hecho, esta solución asume la ejecución de una autenticación EAP completa en cada acceso a un servicio.

IV. Solución propuesta

A. Requisitos

Nuestro trabajo parte de la arquitectura de *eduroam/DAMe* descrita en la sección II-C, con el fin de conseguir extender la federación de identidad y el soporte *cross-layer SSO* no sólo a servicios web, sino a cualquier servicio que se ofrezca en la federación. En concreto, la solución pretende lograr los siguientes requisitos:

1) Requisitos funcionales:

- Acceso federado a los diferentes servicios (web y no-web) disponibles en las instituciones de *eduroam*. El usuario usará las credenciales proporcionadas por su dominio origen para autenticarse.
- Provisión de control de acceso a los servicios mediante Kerberos. Se utilizará la infraestructura AAA existente en eduroam para proporcionar el acceso federado.
- Provisión de *cross-layer* SSO a los usuarios, integrando la solución SSO proporcionada por Kerberos, con la distribución del eduToken proporcionada por eduroam/DAMe.
- La decisión final de acceso estará determinada por la información de identidad del usuario (atributos definidos en su organización origen), y no sólo por su identificador.

2) Requisitos de seguridad:

- Los usuarios en *roaming* deben autenticarse con el KDC en la institución visitada. Esta autenticación se realizará por medio del *eduToken* y del material criptográfico resultante de la autenticación para el acceso a la red. El KDC no necesita tener ningún estado pre-establecido con dicho usuario.
- La información sensible debe distribuirse de forma protegida (ej. identidad y material criptográfico), dado que es susceptible de diferentes ataques, desde *eavesdropping* a *tampering*. Por tanto, se usarán canales protegidos que proporcionen confidencialidad, integridad, autenticación y detección de reenvíos.
- La derivación de material criptográfico se hará de una forma segura, evitando que el compromiso de una clave degrade la seguridad del resto de claves.
- Se protegerá el identificador de usuario mediante el uso de seudónimos, de forma que sólo las entidades de la institución origen conozcan el identificador real del usuario.

[4]http://www.terena.org/activities/tf-emc2/

B. Descripción de la arquitectura

La Fig. 1 muestra la arquitectura propuesta, incluyendo los elementos, las interfaces que los interrelacionan y los protocolos utilizados.

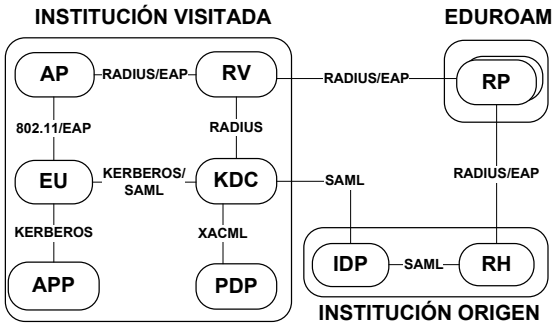

Fig. 1. Descripción de la arquitectura

- *Usuario (EU)*. Representa un usuario de una organización origen que se autentica para acceder a la red en una institución visitada. Además, quiere acceder a los servicios ofrecidos dentro de la misma usando su identidad federada.
- *Servidor RADIUS origen (RH)*. Implementa la funcionalidad de *EAP server*, autenticando al EU durante el acceso a la red. Interactúa con el *IDP* para obtener el *eduToken* y proporcionárselo al usuario. También distribuye material criptográfico al *RV*, que servirá para derivar claves para los servicios de la institución visitada.
- *Proveedor de identidad (IDP)*. Genera y distribuye el *eduToken* al *RH*. También distribuye atributos de usuario al *KDC*.
- *Servidor RADIUS visitado (RV)*. Actúa de proxy RADIUS entre el *AP* y el *RH*. Además, deriva una clave y la distribuye al KDC, para que sirva como secreto compartido de éste con el EU.
- *Punto de acceso (AP)*. Actúa como punto de enlace entre el EU y la red de la institución visitada, interactuando con la infraestructura RADIUS para autenticar al EU.
- *Kerberos Distribution Center (KDC)*. Elemento central de la infraestructura Kerberos. Distribuye tickets al usuario, con los que se realiza el control de acceso a los servicios de la institución. Puede realizar un proceso de autorización basado en los atributos recibidos del *IDP* y en la decisión tomada por el *PDP*.
- *Policy Decision Point (PDP)*. Gestiona las políticas de control de acceso de la institución visitada, y toma

decisiones de autorización basándose en la información recibida del KDC (ej. atributos).
- *Servidor de aplicación (APP)*. Proporciona el servicio específico (ej. servicio HTTP, SSH, FTP...). El acceso al servicio se controla mediante el uso del protocolo Kerberos.

V. Operación general

La solución propuesta en este artículo consta de cuatro fases. Estas fases son necesarias para que un un usuario visitando una institución obtenga acceso a la red y, posteriormente, acceso a los servicios de la misma.

A. Fase 1: Acceso a la red

Antes de que el EU pueda acceder a ningún servicio federado debe autenticarse para acceder a la red. Esta autenticación se lleva a cabo mediante EAP, con el soporte de la infraestructura RADIUS desplegada por eduroam. Este proceso se detalla en la Fig. 2. En concreto, se utiliza un método EAP tunelado (PEAP o TTLS) (1). Así se proporciona privacidad, ya que el identificador real de usuario se transmite protegido, usándose *anonymous@home* como NAI (RFC 4282) público para establecer dicho túnel.

Los paquetes *EAP* generados por el EU son transportados desde el AP hasta el RH mediante la infraestructura RADIUS. El RH los procesa y genera los siguientes paquetes *EAP* para el EU. Este proceso se repite hasta completar la autenticación. Una vez completada, tanto el EU como el RH derivan dos claves: MSK y EMSK. La MSK se envía a través hacia el AP dentro del mensaje RADIUS *Access-Accept*(5 y 6). El AP utiliza la MSK para derivar las claves necesarias para proteger el tráfico con el EU (que puede derivar las mismas claves partiendo de la MSK).

Además, tal y como se define en DAMe [8], una vez que el RH ha verificado la identidad del EU, contacta con el IDP usando el protocolo SAML para obtener el *eduToken* asociado al usuario (3). Este *eduToken* contiene un seudónimo que identifica al EU para futuras interacciones con el IDP, así como información acerca de la autenticación realizada (ej. tiempo de validez de la sesión). El *eduToken* se envía al EU, de modo protegido, a través del túnel establecido por el método EAP (4).

Este trabajo propone extender esta fase, de forma que el primer paquete RADIUS enviado desde el RV hacia el RH contenga una solicitud de *DSRK - Domain-Specific Root Key* (2). Esta clave se deriva de la EMSK siguiendo el proceso de descrito en [14], y sirve como clave raíz para derivar secretos compartidos entre el EU autenticado y los diferentes servicios una institución. También se deriva un *EMSKName* [14], que se utiliza como nuevo identificador de usuario para el mensaje RADIUS *Access-Accept* (5 y 6). Tanto la solicitud de la clave como su posterior distribución al RV (5) se realiza haciendo uso de los atributos RADIUS descritos en [19]. La DSRK se utilizará en la fase 2 para la pre-autenticación del usuario con el KDC.

Al finalizar esta fase, el usuario ha obtenido acceso a la red y tiene el eduToken almacenado de forma local, siguiendo el proceso descrito en DAMe.

B. Fase 2: Pre-autenticación Kerberos y obtención del TGT

En esta fase el EU realiza un proceso de *pre-autenticación* Kerberos con el KDC, basado en el uso del *eduToken* y del material criptográfico derivado de la autenticación de red. En concreto, el EU debe presentar tanto el *eduToken* como el *EMSKName* al KDC. Para transportar esta información se define un nuevo elemento de pre-autenticación llamado *PA-EDUTOKEN*, que contiene además una marca de tiempo para evitar ataques de reenvíos. Dado que tanto el *EMSKName* como el *eduToken* se consideran información sensible (contienen identificadores, tiempos de validez, atributos...), deberá asegurarse que la información es transmitida de forma segura. Para ello se hace uso de FAST [11], que es el mecanismo estándar para proteger los intercambios Kerberos, y en especial, la información de pre-autenticación.

La Fig. 3 muestra el proceso de pre-autenticación. En primer lugar se realiza un intercambio *Anonymous PKINIT* [20] con el KDC (1). Este paso es requerido por FAST para proporcionar un *armor TGT* al EU, que se utiliza para establecer un secreto compartido entre el KDC y el EU, llamado *armor key*.

Tras esto, el EU usa la EMSK generada durante el acceso a la red por el método EAP para derivar el *EMSKName* correspondiente y la DSRK asociada a la institución visitada, tal y como realizó el RH. De la DSRK se deriva una clave intermedia denominada *DSUSRK - Domain-Specific Usage-Specific Root Key* [14], que es específica para el KDC, y de ésta se deriva a su vez la *reply key*, con la que se protegerá el contenido del mensaje KRB_AS_REP, como se especifica en Kerberos [10].

Una vez derivadas las claves, el EU envía el mensaje KRB_AS_REQ que transporta el *PA-EDUTOKEN* (2) al KDC. Este mensaje se protege mediante los mecanismos de FAST descritos en [11]. Además, dado que por omisión el KDC generará un error cuando el identificador del EU incluido en este mensaje no se encuentre en la base de datos (como es el caso de los usuarios en *roaming*), se propone el uso del valor *WELLKNOWN:FEDERATED* como identificador de usuario. Este valor sigue el esquema propuesto en [21], y permite que el KDC reconozca fácilmente a los usuarios en *roaming*, permitiendo la pre-autenticación aunque el usuario no exista en la base de datos.

El KDC valida la firma y la sintaxis del *eduToken* (3) y, usando el valor de *EMSKName* como identificador de usuario, solicita una DSUSRK al RV (4 y 5). Esta solicitud es similar a la solicitud de la DSRK realizada en la fase 1. Una vez obtenida esta clave, el KDC deriva la *reply key* y genera el mensaje KRB_AS_REP, que contiene el TGT para el usuario (6). Partes de este mensaje van protegidas con la *reply key* [10], de forma que sólo el usuario legítimo podrá hacer uso del TGT incluido. Además, el KDC introduce el *eduToken* dentro del campo *authorization_data* del TGT. De esta forma se asegura que éste se recibirá durante la fase 3.

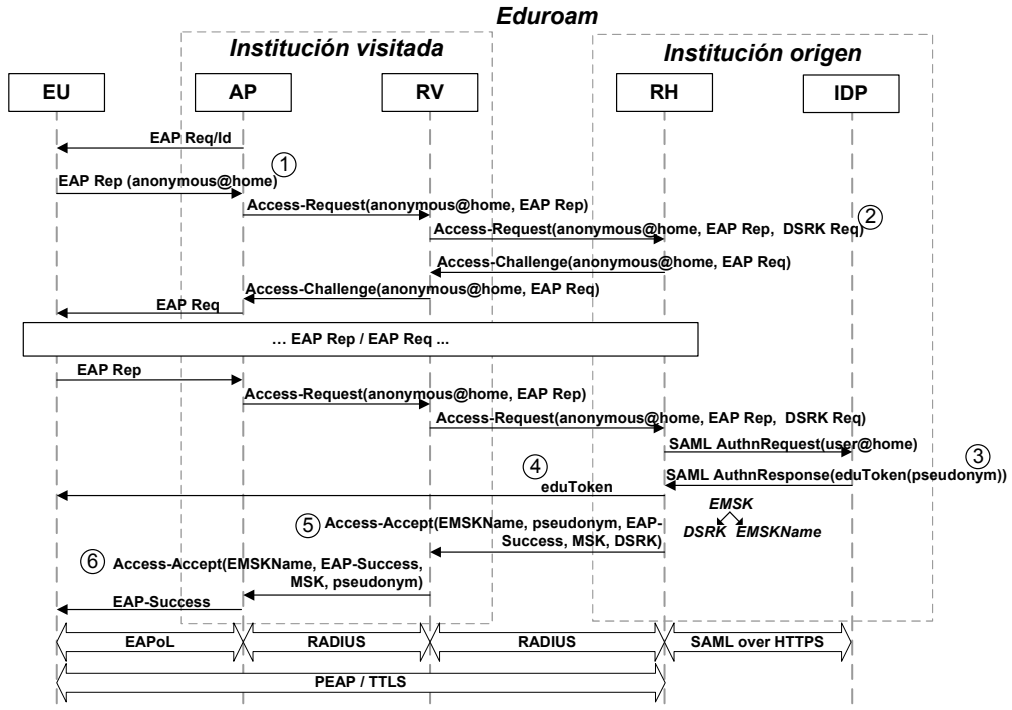

Fig. 2. Autenticación para acceso a la red y distribución del eduToken y material criptográfico

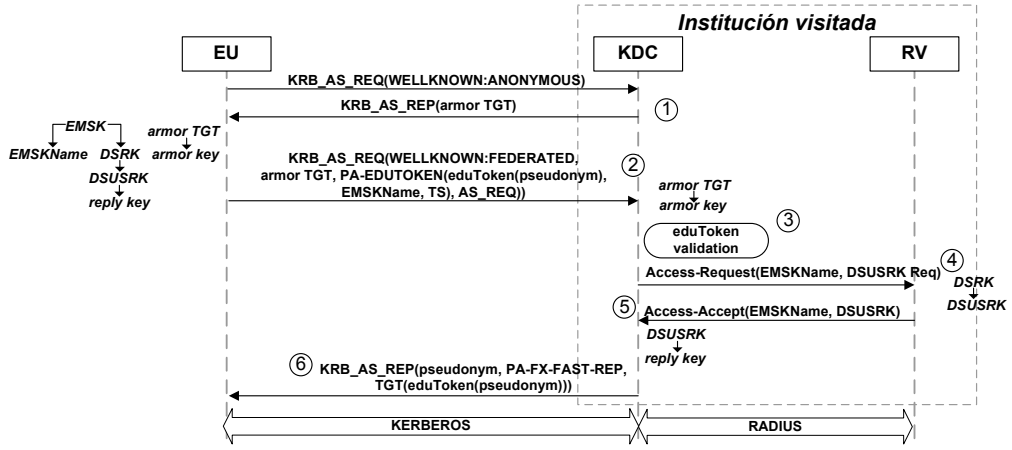

Fig. 3. Pre-autenticación Kerberos y obtención del TGT

C. Fase 3: Obtención del ST y autorización

De manera opcional, el KDC puede realizar un proceso de autorización durante el intercambio TGS, a fin de determinar si el EU cumple los requisitos especificados en las políticas de acceso para el servicio indicado. Este proceso de autorización es similar al detallado en [17], por lo que en este artículo únicamente se proporciona un breve resumen del mismo.

Tal y como se muestra en la Fig 4, el EU comienza el intercambio enviando un mensaje KRB_TGS_REQ al KDC, donde se indica el servicio que se quiere acceder, y se incluye el TGT obtenido en la fase 2 (1) . En primer lugar el KDC valida el TGT recibido, verificando que EU es quien dice ser. Posteriormente, el KDC extrae el *eduToken* del TGT y, usando el seudónimo contenido en el mismo, solicita al IDP los atributos del EU, mediante el protocolo SAML [22] (2 y 3). Este proceso no puede realizarse durante la fase 2 ya que el KDC aún no conoce el servicio final al que quiere acceder el usuario.

Una vez obtenidos los atributos, el KDC contacta con el PDP mediante el protocolo XACML [15], solicitando una decisión de autorización y proporcionando la información de identidad recibida del IDP. El PDP consulta la colección de políticas y toma una decisión (PERMIT o DENY) basándose en las mismas. Si la decisión es positiva, se entrega el ST al EU (6).

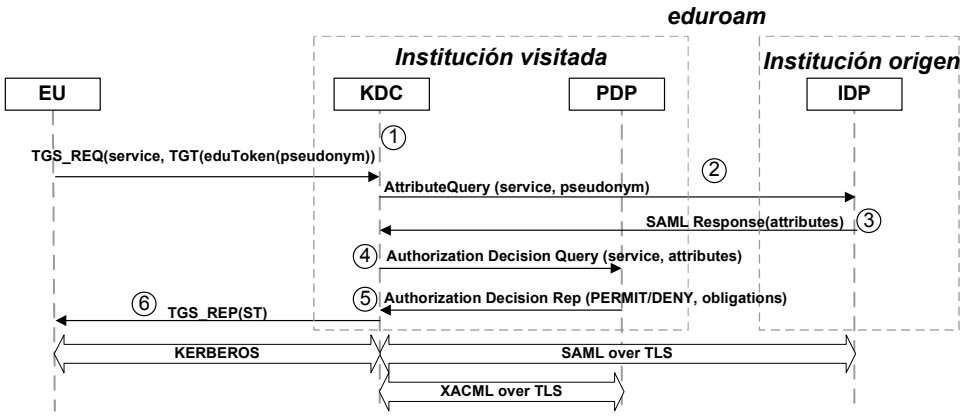

Fig. 4. Autorización y obtención del ST

D. Fase 4: Acceso al servidor de aplicación

Una vez que el EU dispone del ST para acceder al APP, puede seguir el proceso Kerberos estándar con el mismo, ya sea directamente, a través del *GSS-API Kerberos V5 mechanism* (RFC 4121), o a través del *Kerberos V5 SASL mechanism* (RFC 4752), en función de los mecanismos soportados por ambos. Si el ST es válido, el usuario obtendrá finalmente el acceso al servicio.

VI. Conclusiones y Trabajo Futuro

Este artículo describe cómo la red eduroam se puede extender para proporcionar a los usuarios un acceso federados a servicios no-web. Integrando Kerberos en esta infraestructura, los usuarios pueden hacer uso del *eduToken* obtenido durante la autenticación para el acceso a la red y de las claves derivadas de la misma para realizar una pre-autenticación Kerberos y obtener un TGT. Por tanto, la federación de red evoluciona hacia un federación *cross-layer* donde hay cabida para todo tipo de servicios. Además, la distribución del *eduToken* al KDC permite que se realice una autorización avanzada antes de emitir un ticket de servicio al usuario.

Como trabajo futuro, estamos modelando esta arquitectura para su validación formal mediante alguna herramienta de validación de protocolos (ej. AVISPA [23]). Además, estamos realizando la implementación de un prototipo que cubra la funcionalidad básica del mismo.

Agradecimientos

Este trabajo está financiado por el proyecto MULTIGIGABIT EUROPEAN ACADEMIC NETWORK (FP7-INFRASTRUCTURES-2009-1). Los autores agradecen tambień a la Fundación Séneca por el Programa de Ayuda a los Grupos de Excelencia (04552/GERM/06).

References

[1] Assertions and protocol for the OASIS Security Assertion Markup Language (SAML) V1.1, Sept. 2003. OASIS standard.
[2] Openid web site. http://openid.net/. Last access date: 2012/01/19.
[3] OAuth. http://oauth.net. Last access date: 2012/01/19.
[4] K. Wierenga and others. *DJ5.1.4: Inter-NREN Roaming Architecture.Description and Development Items*, September 2006. Project Deliverable.
[5] C. Rigney, S. Willens, A. Rubens, and W. Simpson. *Remote Authentication Dial In User Service (RADIUS)*. IETF RFC 2865, June 2000.
[6] *IEEE 802.11 (2007) Std., Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, June 2007. IEEE Standards.
[7] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. *Extensible Authentication Protocol (EAP)*. RFC3748, June 2004.
[8] *DAMe Project*. http://dame.inf.um.es. Last access date: 2012/01/19.
[9] GEANT Project . http://www.geant.net/pages/home.aspx. Last access date: 2012/01/24.
[10] C. Neuman, T. Yu, S. Hartman, and K. Raeburn. *The Kerberos Network Authentication Service (V5)*. IETF RFC 4120, July 2005.
[11] S. Hartman and L. Zhu. *A Generalized Framework for Kerberos Pre-Authentication*. IETF RFC 6113, April 2011.
[12] F. Butlera, I. Cervesatob, A. D. Jaggardc, A. Scedrovd, and C. Walstadd. *Formal analysis of Kerberos 5*. *Theoretical Computer Science*, 367(1-2):57 – 87, 2006.
[13] B. Aboba, D. Simon, and P. Eronen. *Extensible Authentication Protocol Key Management Framework*. RFC 5247, August 2008.
[14] J. Salowey, L. Dondeti, V. Narayanan, and M. Nakhjiri (2008). *Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)*. RFC 5295, August 2008.
[15] *eXtensible Access Control Markup Language (XACML) Version 2.0*, February 2005. OASIS Standard.
[16] Manuel Sánchez, Gabriel López, Óscar Cánovas, and Antonio F. Gómez-Skarmeta. Performance analysis of a cross-layer sso mechanism for a roaming infrastructure. *J. Netw. Comput. Appl.*, 32:808–823, July 2009.
[17] Rafael Marín-López, Fernando Pereníguez, Gabriel López, and Alejandro Pérez-Méndez. *Providing EAP-based Kerberos pre-authentication and advanced authorization for network federations*. *Computer Standards & Interfaces*, 33(5):494 – 504, 2011.
[18] J. Howlett and S. Hartman. *Project Moonshot*. February 2010.
[19] G. Zorn, T. Zhang, J. Walker, and J. Salowey. *Cisco Vendor-Specific RADIUS Attributes for the Delivery of Keying Material*. April 2011.
[20] L. Zhu and B.Tung. *Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)*. IETF RFC 4556, June 2006.
[21] L. Zhu. *Additional Kerberos Naming Constraints*. IETF RFC 6111, February 2011.
[22] S. Cantor, J. Kemp, R. Philpott, and E. Maler (Eds.). *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) v2.0*, March 2005.
[23] IST AVISPA Project 2001-39252. *Automated Validation of Internet Security Protocols and Applications (AVISPA)*. http://www.avispa-project.org/.