eID Trust Service

Guía para desarrolladores de Software

Rolosa HyJ S.A. - MICITT

13 de Junio de 2014

Resumen

El presente documento es una guía para desarrolladores de software útil como punto de entrada para integrar el eID trust Service en una aplicación web.

Tabla de contenido

Introducción 1

Pre-requisitos 2

Cliente del web service XKMS 2 3

- Depuración 3
- Proxy 3
- WS Security 3
- Autenticación TLS Unilateral 4

Validación de Cadena de Certificados 5

Validación Histórica 6

Certificados TSA 6

Cliente de Alta disponibilidad XKMS2 6

Introducción

El eID Trust Service provee un web service basado en el estándar W3C XKMS v.2.0 [http://www.w3.org/TR/xkms2/] para validación de una cadena de certificados contra un cierto dominio de confianza.

Este web service también permite validación histórica de una cadena de certificados, donde se puede proveer de los datos necesarios de revocación utilizando o una lista de respuestas OCSP y/o CRLs o utilizando el elemento **XAdES RevocationValues.**

Además del web service, también permite la validación de atributos de los certificados y tokens de sellado de tiempo.

Se pueden encontrar más detalles técnicos sobre el web service XKMS2, en el documento de Arquitectura provisto en el paquete SDK.

Pre-requisitos

Para proseguir con el presente documento, es necesario disponer de:

- Una Tarjeta Inteligente para Firma Digital de Costa Rica
- Un lector de tarjetas apropiado
- Certificados Digitales de la cadena de la confianza de la Firma Digital de Costa Rica
- Drivers necesarios para la Tarjeta Inteligente, dependiendo el sistema operativo en el que se desee realizar los trabajos.

Los drivers y certificados digitales tienen que ser obtenidos desde el sitio de Soporte de Firma Digital de Costa Rica:

https://www.soportefirmadigital.com/sfd/default.aspx

Cliente del web service XKMS 2

Para acceder al web service XKMS v.2.0, el SDK provee un componente XKMS2Client . Para inicializar este componente, se toma un argumento de ubicación en forma de String, el cual espera ser la ruta completa del XKMS2 web service, ej. http://www.alpha.rolosa.com:8443/eid-trust-service-ws/xkms2

Depuración

Durante el proceso de desarrollo, se puede requerir mayor información en el envío y la recepción de mensajes SOAP. Se puede habilitar esta característica utilizando el método **setLogging(boolean)**

Proxy

Si se está trabajando detrás de un proxy, se puede configurar esto en el XKMS2Client utilizando el método **setProxy (String proxyHost, int proxyPort)**

WS Security

El Trust Service al que se esté conectando puede estar configurado para utilizar WS-Security para firmar los mensajes de salida. En este caso, el cliente XKMS2 verificara automáticamente la firma del mensaje de entrada utilizando el certificado incrustado en el elemento WS-Security. Opcionalmente se puede agregar mayor verificación para este certificado especificando manualmente el certificado que se espera, utilizando el método **setServerCertificate(X509Certificate)**.

Por defecto el tiempo (timestamp) actual del WS-Security será verificado que no tenga una diferencia mayor a 5 minutos. Se puede configurar este parámetro utilizando el método **setMaxWSSecurityTimestampOffset(long)** (en segundos).

Autenticación TLS Unilateral

Cuando se accede al Trust Service a través de SSL, se puede especificar opcionalmente la Clave Publica a ser usada para la verificación del certificado SSL utilizando el método **setServicePublicKey(PublicKey)**

Validación de Cadena de Certificados

En su mínima configuración, un cliente puede especificar una ruta de certificados para su validación. El Trust service realizara una validación-PKI en contra de el dominio de confianza configurado por defecto. Se espera que la ruta de certificados contenga el certificado raíz como el último elemento de la lista.

En el caso de que la validación no fuera exitosa, una **ValidationFailedException** será lanzada, conteniendo la lista de **XKMS2 reason URIs** [http://www.w3.org/TR/xkms2/#XKMS_2_0_Section_5_1] que causaron que la validación-PKI falle. Estas razones-URI pueden también ser obtenidas utilizando el método **getInvalidReasons()**

En el caso de que se requiera validar contra un dominio de confianza especifico, el API del cliente XKMS2 permite especificar esto. Opcionalmente se puede requerir que el Trust Service retorne los datos de revocación (respuestas OCSP y/o datos CRL). Estos datos de revocación pueden ser obtenidos después, utilizando el método el cual retorna un elemento **XADES revocation values.**

Validación Histórica

Es posible realizar validación histórica en rutas de certificados. El cliente XKMS2 provee 3 formas de proveer los datos necesarios de revocación para el Trust Service, ya sea especificando la lista de OCSPResp y X509CRL, o especificando la lista de respuestas OCSP codificadas y datos CRL, o alternativamente especificando explícitamente el elemento **XAdES revocation values** [http://www.w3.org/TR/XAdES/#Syntax_for_XAdES-L_form_The_RevocationValues_element].

Validaciones históricas son útiles en el contexto de la verificación de firmas digitales ETSI XAdES v1.3.2 , enfocadas en las firmas XAdES-X-L.

Certificados TSA

Es posible validar certificados TSA utilizando el web service XKMS2. El toquen de sellado de tiempo necesitara contener la ruta de certificados del certificado TSA para que el Trust Service pueda realizar la validación-PKI.

Se puede especificar el token de sellado de tiempo utilizando la clase de BouncyCastle **TimeStampToken.**

Cliente de Alta disponibilidad XKMS2

El **HAXKMS2Client** es un componente de alta disponibilidad que provee las mismas características que el **XKMS2Client**, pero permite especificar un validador jTurst personalizado **TrustValidator**, para ser utilizado en caso que la solicitud de validación que se realiza al web service XKMS falle inesperadamente debido a problemas de conectividad. En ese caso se utilizara el TrustValidator especificado.