Infraestructura de clave pública en una Universidad del Paraguay

Samuel Cantero, Benjamín Barán, Felipe Stuardo

scanterog@gmail.com bbaran@pol.una.py fstuardo@uca.edu.py

Universidad Católica Nuestra Señora de la Asunción

Abstract: *This document presents a comparative analysis among open source software packages with the largest number of references for PKI management, which ultimately results in the use of EJBCA as the solution for the case of study: the Catholic University of Asuncion. The implementation of the selected tool presents different problems, which derive mainly from the intrinsic nature of the EJBCA architecture, since the application integrates the functions of registration and emission of certificates. For this reason, this work proposes and implements the separation of such functions through the construction of a registering application, which is capable of integrating itself with EJBCA, providing a more general PKI solution. Finally, the implementation of a new general solution is presented for the Catholic University of Asuncion.*

Keywords: Cryptography, Digital Certificate, PKI, EJBCA

1. Introducción

En la actualidad, millones de personas utilizan las redes para sus transacciones bancarias, compras, declaraciones de impuestos, entre otras actividades y muchas de estas actividades son críticas, ya que involucran información confidencial [1].

Esto genera la necesidad de proveer un método que permita identificar la autenticidad de las partes en una transacción de manera a evitar la suplantación de identidad. También genera la necesidad de proveer canales seguros para las transacciones de manera a evitar el robo de información.

Considerando el aumento de amenazas sobre la seguridad de la información y de los sistemas, es necesario implementar una infraestructura que sea capaz de salvaguardar toda la información sensible dentro de una institución. Por ejemplo, para las transacciones en línea o el comercio electrónico, es ineludible la alta confiabilidad y el uso de una infraestructura segura. Está claro que debemos contemplar mecanismos que garanticen la seguridad.

En la actualidad, las técnicas y algoritmos criptográficos constituyen la base de la seguridad computacional y de las redes de computadoras, los cuales permiten proveer servicios de seguridad incluyendo autenticación, confidencialidad, integridad y no repudio [1].

Los algoritmos criptográficos permiten realizar cifrado de información, firma de documentos, control de autenticación de personas, control de integridad de documentos y más. El problema con estos mecanismos radica en el modo de operación, principalmente porque debemos lidiar con el robo de información, la suplantación de identidad, entre otros tipos de estafas [2].

Los principales algoritmos criptográficos utilizados son los algoritmos simétricos y los algoritmos de clave pública. En la criptografía de clave simétrica, se requiere que los usuarios compartan una clave secreta común para lograr una comunicación segura. El problema está en la manera de intercambiar esa clave común entre personas que no se conocen o se encuentran a miles de kilómetros de distancia. En la criptografía de clave pública, cada usuario tiene un par de claves, una clave pública y una clave privada, y se requiere que los usuarios intercambien sus correspondientes claves públicas para lograr una comunicación segura. El problema está en cómo identificar que una determinada clave pública le pertenece a la persona correcta.

Por esta razón, para lograr el intercambio seguro de claves, se recurre a una tercera parte de confianza que certifique que una clave pública corresponde a una determinada entidad (persona física, persona jurídica, o dispositivo). Esta certificación se realiza a través de los certificados digitales. Para lograr un entorno seguro de administración de claves públicas a través de una tercera parte de confianza, se necesita una infraestructura que gestione el ciclo de vida de los certificados digitales. Esta infraestructura se conoce como Infraestructura de Clave Pública (*Public Key Infrastructure* – PKI, en inglés).

2. Análisis de soluciones

Todo el marco teórico necesario para un entorno de PKI (criptografía, firma digital, certificados digitales, etc.) se puede encontrar en [3].

Para llevar a cabo la implementación de la infraestructura de clave pública en la Universidad Católica Nuestra Señora de la Asunción, primero se han analizado varios paquetes de *software open source* que permitan gestionar una solución.

Entre los paquetes de *software open source* más referenciados en el proceso de investigación para la gestión de una infraestructura de clave pública se encontraron los siguientes [4]: EJBCA (*Enterprise Java Bean Certificate Authority*), OpenCA PKI, OpenXPKI, y *Dogtag Certificate System*.

De los paquetes listados, se descartó para el análisis comparativo a *Dogtag Certificate System* por no contar con soporte multiplataforma y estar limitado únicamente a una de las familias de distribución Red Hat: Fedora [5].

	EJBCA	OpenCA	OpenXPKI
Inicios	2001	1998	2008
Implementación	Java, J2EE	C, Perl, JS, PLSQL, Unix Shell	Perl
Sistemas Operativos	Independiente del SO	Linux, MAC OS X, Solaris, BSD	FreeBSD, Linux, Solaris, OpenSolaris, Mac OS X
Base de datos	MySQL, PostgreSQL, Oracle	MySQL, PostgreSQL, Oracle	MySQL, PostgreSQL, Oracle
Soporte LDAP	Sí	Sí	Sí
Soporte OCSP	Sí	Sí	No
Escalabilidad	Escalable	No tan escalable	-
Configuración	Demasiado compleja	Compleja	Complicado
Licencia	LGPL	BSD revisada	Apache 2.0
Movimiento comunidad	Constante	Lapsos largos de tiempo	Poco movimiento
Último release	4.0.16 el 27-06-2013	1.1.0 el 31-10-2010	2008 (funcional-no estable). En marzo 2012 salió OpenXPKI 0.10.0. Se espera la versión 1.0 desde finales del 2013.
Libros en Amazon	4 libros, 1 énfasis total	3 libros, 1 énfasis total	Ninguno
Casos de éxito	- Ministerio de Defensa Francia; - Ministerio de Finanzas, Francia; - Junta de Policía Nacional Sueca; - Grupo Safa, España; - Serasa, Brasil	- e-Science-UNPL, Argentina; - UTPL, Ecuador; - UNA, Paraguay; - UDB, San Salvador	- Cynops, Alemania

Tabla 1. Análisis comparativo de paquetes de *software* de gestión de PKI.

Por lo tanto, se redujo el análisis a los primeros tres paquetes de *software* arriba listados.

En la tabla 1, se puede observar un análisis comparativo de los paquetes de *software* mencionados. Para la realización del cuadro, se tuvo en cuenta un análisis hecho en Rumania entre OpenCA y EJBCA [6]. Además, se añadió al análisis a OpenXPKI y se insertaron otras variables significativas para el estudio. El análisis muestra una notable ventaja a favor de EJBCA que supera a los otros dos paquetes de *software* en los siguientes aspectos:

1. Su constante evolución, tanto en el ciclo de lanzamientos de nuevas versiones estables, así como de características funcionales en el *software*. En los primeros 6 meses del año 2012, EJBCA ha lanzado dos versiones estables del producto. Por su parte, OpenCA no ha lanzado ninguna nueva versión estable desde el año 2010, y OpenXPKI luego de haber dejado una versión funcional pero no estable en 2008, solo ha retomado el proyecto en marzo de 2012 aproximadamente, esperando lanzar su versión 1.0 a finales de 2013.

2. Cuenta con un soporte mucho más preparado. La comunidad cuenta con una mayor cantidad de usuarios activos y una documentación mucho más extensa y elaborada. La página oficial del proyecto se actualiza constantemente. En cambio, OpenCA y OpenXPKI tienen la página del proyecto desactualizada, soporte reducido y una comunidad muy inferior a EJBCA.

3. Casos de éxitos de instalaciones en instituciones con alto grado de responsabilidad. Los casos de éxito de instalaciones de EJBCA tienen mayor peso que las de OpenCA, que en su mayoría son instalaciones del producto para una red de prueba. OpenXPKI cuenta únicamente con una instalación en Cynops, empresa alemana consultora y especialista en OpenXPKI.

4. Su mayor escalabilidad [6], muy importante si se desea implementar una PKI en constante crecimiento.

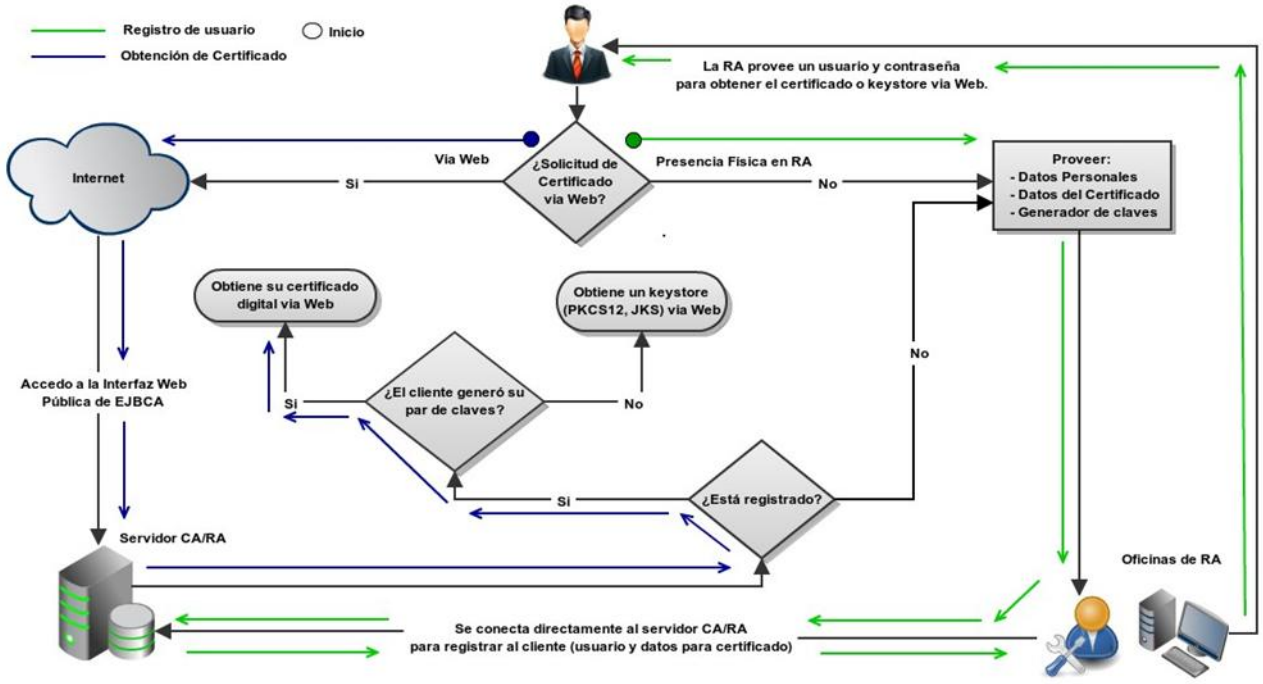

Figura 1. Funcionamiento de EJBCA.

Entre otras ventajas importantes, se encuentran también su adaptación a las recomendaciones del RFC5280 y el uso de una licencia LGPL (*GNU Lesser General Public License*) que permite ser integrada casi sin ninguna limitación con cualquier programa propietario [7]. Por estas razones, se optó por EJBCA como herramienta de gestión de PKI para la Universidad Católica.

2.1. EJBCA como solución de PKI

Antes de realizar la implementación para un entorno de producción en la Universidad, se realizó una instalación de prueba de EJBCA en su versión 4.0.12.

En este entorno de *testing* y evaluación de la herramienta se encontraron algunos problemas en la forma de operar de EJBCA. Antes que nada, es importante mencionar que la instalación de EJBCA por defecto opera simultáneamente como autoridad de certificación (CA) y como autoridad de registro (RA). Para comprender cómo opera EJBCA se puede observar el diagrama de la Figura 1, donde se muestran los pasos que debe realizar un usuario para solicitar su certificado digital. Inicialmente el usuario debe presentarse de forma física a la RA (registro de usuario - flujo verde) donde debe proveer sus datos. Allí un operador de la RA se conecta al servidor de la CA/RA (EJBCA) para registrar al usuario en el sistema a través de un nombre de usuario (*username*) y una contraseña que posteriormente será entregado al usuario para que pueda obtener su certificado digital o *keystore software* (almacén de claves y certificados). También en ese mismo acto el operador de la RA se encarga de registrar los datos que aparecerán en el certificado. Es importante resaltar que no se registran los datos personales del usuario ya que el sistema no cuenta con esa capacidad. Una vez entregado el *username* y la contraseña al usuario, este procede a obtener su certificado digital (obtención de certificado - flujo azul). Para ello ingresa a la interfaz web pública en el servidor de la CA/RA y utilizando el *username* y *password* (condición de registro) proveído por el operador de la RA, se dispone a obtener su certificado digital o *keystore software*. Si el usuario generó su par de claves, entonces deberá proveer a la interfaz web pública de EJBCA un *Certificate Signing Request* (CSR)[27] que contenga su clave pública de manera a que el servidor de la CA le retorne su certificado digital (caso indicado en la Figura 1). En caso contrario, el usuario deberá solicitar la generación de un *keystore software*; fichero con su clave privada y certificado digital.

Se encontraron los siguientes problemas y limitaciones:

1. No hay separación entre los componentes CA y RA de la PKI. Esto genera un problema de seguridad ya que se expone a la CA al permitir que los usuarios accedan directamente al servidor CA (EJBCA) para obtener su certificado digital.

2. La interfaz web pública de EJBCA únicamente permite generar el certificado digital o *keystore software* vía web. En caso de generar un *keystore software* a través de la interfaz web pública, éste viaja por Internet hasta llegar al usuario, exponiendo la clave privada.

3. El operador de la autoridad de registro (RA) se conecta directamente al servidor de la CA (EJBCA) para registrar a los usuarios. Esto permite que cualquier operador de la RA tenga acceso al servidor de la CA generando otro potencial hueco de seguridad en el sistema.

4. El operador de la autoridad de registro sólo tiene la posibilidad de registrar información del certificado, por lo que no se registran informaciones personales de los usuarios. Estas informaciones son muy relevantes para una autoridad de registro.

[27] Un CSR es una petición de certificado que consiste de un nombre distinguido (DN) y una clave pública [8].

5. El usuario no tiene la posibilidad de registrar sus datos personales vía web para la petición de los certificados digitales. Con esta funcionalidad se podría acelerar el proceso de registro y validación de datos de los usuarios. Además, si existe una manera totalmente confiable de validar la identidad del usuario, y en caso de que haya adjuntado su clave pública para la petición del certificado digital, se podrá emitir directamente el certificado al usuario sin necesidad de que el interesado se tenga que presentar personalmente a las oficinas de la RA.

6. La renovación y revocación de los certificados digitales únicamente se realizan de manera presencial en las oficinas de la RA. Se podrían facilitar estas operaciones a los usuarios a través de una solución alternativa vía web. Cabe mencionar que la suspensión de los certificados digitales únicamente se realiza por parte de los operadores de la RA, ya que generalmente implica una sanción hacia el usuario, por ejemplo por falta de pago.

EJBCA trae un módulo adicional denominado *External RA* que permite resolver parcialmente el problema de la separación de los componentes CA y RA (Problema 1), sin embargo este módulo no resuelve los problemas 2 al 6 listados anteriormente. Esto se debe a la naturaleza intrínseca de la arquitectura de EJBCA que tiene muy integrada la función de registro y la función de emisión de certificados.

2.2. Solución a los problemas de EJBCA

De manera a resolver los problemas presentados en la sección anterior, se desarrolló una aplicación RA con las siguientes características [3]:

(a) La aplicación RA se conecta a la aplicación CA (EJBCA) a través de *web services* con autenticación de certificado cliente *Secure Sockets Layer* (SSL). De esta forma ambas aplicaciones interactúan asegurando la autenticidad de la comunicación a través de certificados digitales.

(b) La aplicación RA gestiona el registro de usuarios permitiendo mantener información personal y del certificado del usuario. De ésta manera, el operador de la RA se conecta al servidor de la RA y no al servidor de la CA para la gestión de usuarios.

(c) La aplicación RA permite el registro de datos de usuarios vía web.

(d) La aplicación RA permite la renovación y revocación de certificados vía web.

(e) La aplicación RA permite descargar sus certificados digitales a los usuarios.

El punto (a) permite resolver el Problema 1 referente a la separación de los componentes CA/RA. El punto (b) permite resolver el Problema 3 referente a la conexión directa del operador de RA al servidor de CA para la gestión de usuarios, y también permite resolver el Problema 4 referente a la incapacidad de almacenar información personal del usuario. El punto (c) permite resolver el Problema 5 referente a la incapacidad de registro de datos de usuarios vía web. De esta forma, una entidad final puede solicitar su certificado digital vía web ya que los datos introducidos en el formulario web pueden ser verificados por el operador de la RA. El punto (d) permite resolver el Problema 6 referente a la incapacidad de realizar una renovación o revocación vía web. El punto (e) permite al usuario obtener su certificado digital desde la interfaz RA.

La solución ideal al *Problema 2*, referente a la exposición de la clave privada al viajar el *keystore software* por Internet, consiste en la entrega personal de las claves privadas a los usuarios en un *token hardware*. Sin embargo, esta solución resulta inviable por su alto coste en ciertos entornos, como el universitario, donde se tiene una cantidad elevada de alumnos, funcionarios y profesores. Por lo tanto, se propone una solución alternativa que consiste en la construcción de una PKI con dos clases de autoridad de certificación (CA); la CA clase 1 y la CA clase 2. La diferencia entre estas dos clases radica en el grado de sensibilidad de la aplicación. La *CA clase 1* será la encargada de emitir *tokens software*, los cuales serán entregados personalmente a los usuarios y por lo tanto las claves privadas no serán expuestas a viajar por Internet. Estos *tokens software* serán utilizados para aplicaciones menos sensibles como por ejemplo aplicaciones informativas y académicas. La *CA clase 2* será la encargada de emitir *tokens hardware*, los cuales serán entregados personalmente al usuario. Estos *tokens hardware* serán utilizados para aplicaciones más sensibles como por ejemplo las aplicaciones comerciales y bancarias. En el presente trabajo se propone esta solución alternativa, pero su implementación no se lleva a cabo. Para que esto sea posible, se deberá añadir la funcionalidad necesaria para trabajar con *tokens hardware* a la aplicación RA, lo cual se deja para trabajo futuro de desarrollo e investigación, por el coste que implica.

Para suministrar los certificados y CRL a las entidades finales, se implementó un servidor repositorio HTTP de certificados y CRL. Se desarrollaron *scripts* personalizados para que realicen la tarea de exportar los certificados y CRL generados en la CA al repositorio público. Este repositorio solo contiene los certificados y las CRL en un servidor web sin ninguna interfaz informativa. En la Tabla 2, se muestra la solución propuesta para cada problema citado anteriormente.

2.3. Diagrama de flujo de la solución

En la Figura 2, se puede observar el diagrama de flujo de la solución propuesta para el registro de usuarios de la PKI. Un usuario que desea solicitar su certificado digital puede hacerlo de dos maneras: vía web o de forma física en la autoridad de registro (RA). En el caso que desee hacerlo vía web (flujo azul), deberá acceder a la nueva aplicación RA y registrar sus datos en un formulario web. Al finalizar el registro, la aplicación RA le solicitara validar la dirección de correo electrónico ingresada, de manera a evitar registro de correos de los cuales el usuario no es propietario.

ID del problema	Descripción	Solución
Problema 1	No hay separación de los componentes CA y RA.	La CA (aplicación EJBCA) y la RA (aplicación desarrollada) se encuentran en servidores diferentes.
Problema 2	Exposición de la clave privada del usuario al viajar el *keystore software* por Internet.	Entrega personal del *token* al usuario. Dos niveles de CA para la emisión de *tokens*.
Problema 3	Conexión directa del operador de RA al servidor de CA para la gestión de usuarios.	El operador de la RA se conecta al servidor de RA, que contiene a la aplicación RA, para la gestión de usuarios.
Problema 4	Incapacidad de almacenamiento de información del usuario (personal e institucional).	La aplicación RA permite almacenar información personal de los usuarios, así como relaciones laborales con empresas.
Problema 5	Incapacidad de registro de datos de usuarios vía web para la petición de certificados digitales.	La aplicación RA permite el registro de datos de usuarios vía web para la petición de certificados digitales.
Problema 6	Incapacidad de realizar una renovación o revocación del certificado digital vía web.	La aplicación RA permite la renovación y revocación de certificados vía web.

Tabla 2. Soluciones propuestas a los problemas encontrados en EJBCA.

Una vez validado el correo electrónico, la solicitud del usuario se encontrará en un estado pendiente de aprobación. Aquí finaliza la interacción del usuario con la aplicación RA. A partir de este momento, el usuario deberá esperar que sus datos sean validados por los operadores de la autoridad de registro.

En el caso que un usuario desee solicitar su certificado digital de forma física a la autoridad de registro (flujo verde), deberá acercarse a ella y proveer los datos necesarios. Allí un operador RA (con privilegios de administrador) se conectará a la aplicación RA para registrar sus datos. Una vez que los datos sean validados, se procederá a la emisión del certificado digital.

Obsérvese que el operador RA no se conecta al servidor de CA para administrar el registro de usuarios. Ahora el operador RA se conecta a la aplicación RA ubicado en el servidor de RA, el cual es un servidor independiente del servidor de CA. Una vez que el usuario ha finalizado la interacción con la aplicación RA, el operador RA deberá validar la identidad del usuario y emitir su certificado digital. Este proceso se muestra en el diagrama de flujo de la Figura 3. El operador RA se conecta con privilegios de administrador a la aplicación RA de manera a poder gestionar los registros de usuarios y la emisión de los certificados digitales. Lo primero que realiza el operador RA es verificar los datos proporcionados por el usuario de manera a comprobar su identidad. Una vez que la identidad del usuario es comprobada, el operador RA aprueba la solicitud del usuario, pero en caso de que la identidad del usuario no pueda ser comprobada con los datos registrados, entonces el operador RA rechaza la solicitud del usuario. Si la solicitud es aprobada, el operador RA deberá solicitar la generación del certificado digital al servidor de CA. En caso de que el usuario haya adjuntado su CSR, que contiene el DN[28] y su clave pública, entonces el servidor CA generará el certificado digital y lo publicará en el repositorio. También se enviará una copia del certificado digital al correo electrónico registrado por el usuario. En caso de que el usuario haya solicitado la generación de su par de claves, entonces el servidor CA generará el par de claves y retornará a la aplicación RA un *keystore* con la clave pública y privada del usuario. La clave que protege el *keystore* será entregada al usuario vía correo electrónico. De esta manera, el operador RA solo posee el *keystore,* pero no la clave que permite hacer uso del mismo. También con esto se logra que la clave privada no viaje a través de Internet.

2.4. Implementación de la aplicación RA

Debido a la naturaleza intrínseca de la arquitectura de EJBCA, la cual tiene muy integrada en la misma aplicación EJB (*Enterprise JavaBeans*) la función de registro y la función de emisión de certificados, toda la solución propuesta se basa en la separación de estas funciones a través de la construcción de una nueva aplicación que se encargue de la función de registro. El desarrollo de esta aplicación RA se realizó utilizando las siguientes herramientas: *Framework* Tekoporu [10], JBoss AS [11] y PostgreSQL [12].

Lo primero que se realizó es el diseño del modelo relacional de la base de datos para la aplicación. Este modelo define el conjunto de tablas del sistema junto con sus relaciones. El modelo de datos de la aplicación está construido para operar en un entorno tanto universitario como empresarial. De manera a organizar los datos en grupos lógicos, la base de datos utiliza tres esquemas (*schemas*). Los esquemas utilizados son: *public*, *sysra* y *seguridad*.

El esquema *public* contiene las tablas que serán utilizadas para almacenar de forma temporal los registros de usuarios vía web.

[28] DN (*Distinguished Name*): describe un nombre jerárquico compuesto de atributos [9].

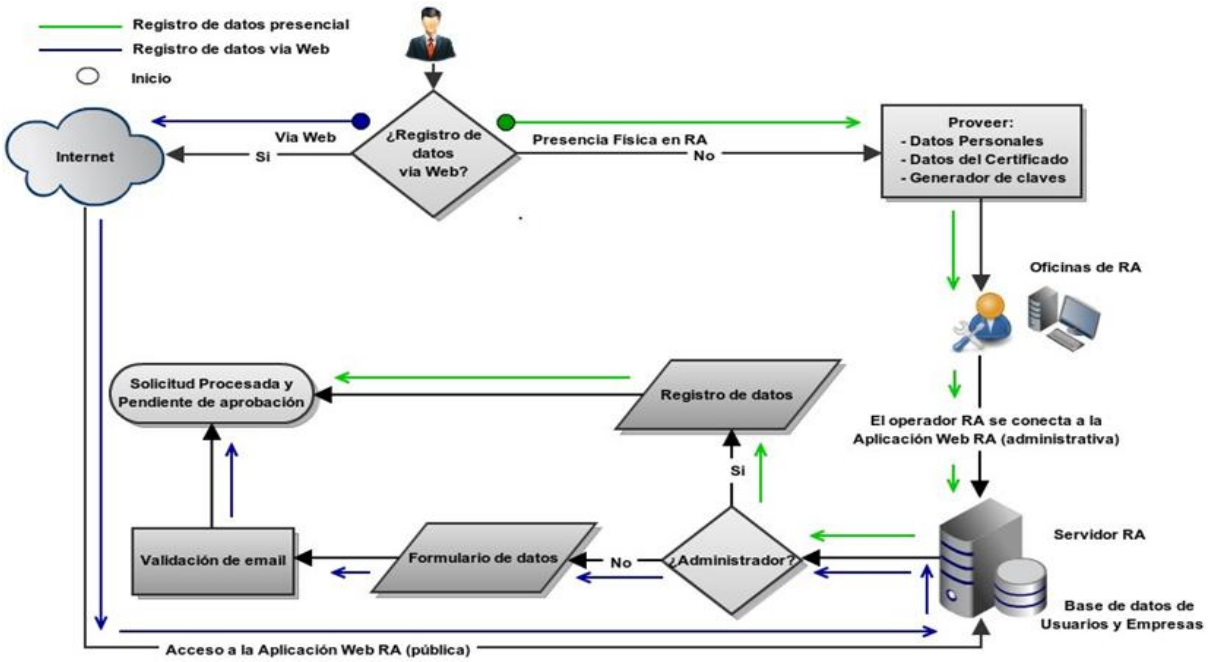

Figura 2. Solución propuesta para el registro de usuarios de la PKI.

El esquema *sysra* contiene las tablas que serán utilizadas para almacenar de forma permanente los registros de usuarios y empresas; también contiene otras tablas importantes que permiten relacionar la aplicación RA con EJBCA. El esquema *seguridad* contiene las tablas que serán utilizadas para almacenar los usuarios de la aplicación RA, es decir, los operadores de la aplicación RA con sus diferentes roles. El rol define las funcionalidades a las cuales tendrá acceso el operador.

Una vez definido el modelo relacional de la base de datos, se definieron los módulos de la aplicación. La aplicación RA tiene módulos para la gestión de personas y empresas, solicitudes nuevas de certificados, solicitudes aprobadas, usuarios (operadores) de la aplicación RA, roles y funcionalidades de usuarios, datos universitarios específicos como: facultades, departamentos, carreras y cargos. También tiene un módulo para la generación, revocación, renovación y descarga de certificados.

Otra funcionalidad importante de la aplicación RA construida es el método de autenticación al sistema. La aplicación RA construida utiliza una doble autenticación. Para poder acceder a la tradicional ventana de inicio de sesión usuario-contraseña, el operador primero deberá autenticarse con un certificado digital al servidor y a la aplicación RA. Este método de autenticación permite asegurar que sólo el operador que tenga el certificado cliente adecuado podrá acceder a la ventana de inicio de sesión usuario-contraseña. No se utiliza solo la autenticación por certificado cliente, ya que en caso de que el certificado sea robado por un atacante, éste solo podrá acceder a la ventana de inicio de sesión usuario-contraseña, siendo la misma otra barrera adicional que el atacante deberá superar.

Integración de la aplicación RA con EJBCA

Como la funcionalidad de registro se encuentra en la aplicación RA y la funcionalidad de emisión de certificados se encuentra en la aplicación EJBCA, es necesario lograr que ambas herramientas interactúen entre sí para lograr el objetivo deseado en la infraestructura de clave pública.

Para poder comprender cómo se integran ambas aplicaciones, es necesario mencionar que la aplicación RA almacena todas las solicitudes de usuario para la generación de sus certificados en una tabla llamada *solicitud_ejbca*. Esta tabla contiene información del solicitante e información necesaria para la generación del certificado digital, y un campo muy importante para la integración de ambas herramientas, denominado *username*. Sabiendo que EJBCA registra a sus usuarios a través de un nombre de usuario, único para cada uno, y una contraseña, la aplicación RA registra a los usuarios en la aplicación EJBCA con el valor definido para cada solicitud en el campo *username*. Este campo *username* de la aplicación RA, mapeado con el de EJBCA, está formado por un prefijo definido para la aplicación RA concatenado con el ID de la clave primaria de la tabla *solicitud_ejbca*. Por ejemplo, si el prefijo para la aplicación RA es **AREG** entonces la primera solicitud se registraría en EJBCA con el *username* **AREG1**, la segunda con **AREG2**, y así sucesivamente.

Cómo una PKI puede tener varias autoridades de registro, se define un prefijo único para cada instancia de la aplicación RA. Por ejemplo, si tenemos la facultad de ingeniería y la facultad de contables dentro de una universidad y deseamos una RA dentro de cada facultad, podemos identificar a la facultad de ingeniería con el prefijo *ING* y la facultad de contables con el prefijo *CONT*. Esto permitirá que los usuarios de la facultad de ingeniería se registren en EJBCA con un *username* distinto al de los usuarios de la facultad de contables. Por ejemplo, **ING1** para el primer usuario de la facultad de ingeniería y **CONT1** para el primer usuario de la facultad de contables.

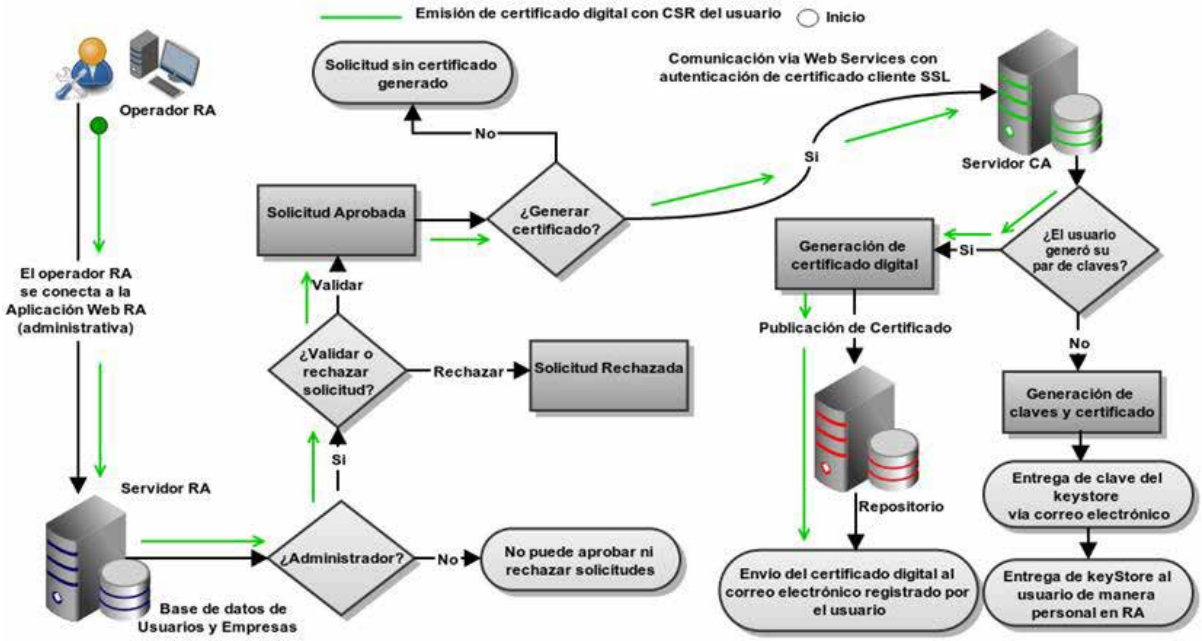

Figura 3. Solución propuesta para la validación de datos y para la emisión de certificado digitales.

Para que la aplicación RA pueda registrar a los usuarios en la aplicación EJBCA, utiliza los *web services* proveídos por EJBCA. Para poder consumir estos servicios web, EJBCA debe proveer un certificado digital SSL con los permisos necesarios a la aplicación RA. Este certificado digital permite a la aplicación RA tener acceso únicamente a las siguientes funcionalidades: registro de usuarios, generación de certificados, revocación de certificados, renovación de certificados, y obtención de certificados. Es decir, con este certificado digital la aplicación RA no podrá modificar perfiles de certificados ni perfiles de entidades finales. Tampoco podrá expedir otros certificados para operadores de RA, entre otras funcionalidades importantes de EJBCA. Este certificado especial es asignado en EJBCA con los permisos mencionados a través de su número de serie, que junto con el emisor del certificado constituyen un par único. De esta manera, se garantiza que no podrá existir otro certificado digital que pueda tener acceso a las funcionalidades citadas. En caso, que se tenga varias RA, EJBCA podrá emitir un certificado digital SSL para cada RA.

3. Implementación de una PKI

Para la implementación de la infraestructura de clave pública en el caso de estudio: el campus universitario de Santa Librada de la Universidad Católica "Nuestra Señora de la Asunción" - UCA, se han seguido los siguientes pasos: definición del diagrama de red; instalación y configuración de la CA Raíz (EJBCA); instalación y configuración de las CA Subordinadas (EJBCA); instalación y configuración de la aplicación RA; instalación y configuración del repositorio.

Como quedó fuera del alcance del proyecto la validación del Ministerio de Industria y Comercio - MIC (institución designada por la Ley 4610/12 como Autoridad de Certificación Raíz del Paraguay) a la UCA como autoridad certificadora de segundo nivel dentro de la PKI del Paraguay, se construyó una CA raíz para la UCA (CA raíz UCA) con dos CAs subordinadas a ella: la CA clase 1 y la CA prueba. La *CA clase 1* emite los certificados digitales y claves privadas en *tokens software* para ser utilizadas en aplicaciones académicas e informativas, mientras que la *CA Prueba* es utilizada para testear el funcionamiento de la infraestructura y educar a los potenciales usuarios.

Para el diagrama de red, se consideraron ciertos factores de seguridad importantes para un entorno de PKI, tales como:

- El aislamiento del servidor de la autoridad de certificación raíz del resto de las estaciones de trabajo. Esto es crítico, ya que la clave privada de la CA raíz no debe estar expuesta a ataques de manera a garantizar toda la seguridad de la infraestructura.
- La restricción en el acceso a las autoridades de certificación subordinadas encargadas de emitir los certificados digitales a las entidades finales. El acceso a las CA subordinadas debe estar permitido únicamente a la aplicación de la autoridad de registro.
- La ubicación de la/s autoridad/es de registro y del repositorio público en la red DMZ (*DeMilitarized Zone*). Esto permite separar la red de servidores públicos de la red local.

Para la implementación en producción de la PKI en la Universidad Católica Nuestra Señora de la Asunción se utilizaron las siguientes herramientas: Citrix XenServer 6 [13], CentOS 6.4 x86 64 [14], JBoss AS 5.1.0.GA [11] y PostgreSQL 9.2 [12].

Los tipos de certificados emitidos por la PKI son los siguientes: certificados para personas (alumnos, profesores y funcionarios); certificados para servidores web, y certificados para servidores de correo electrónico.

Resultados de la implementación

Los usuarios (alumnos, profesores y funcionarios) pueden solicitar su certificado digital vía web. Estos certificados

digitales permiten la firma digital de documentos en formato *Word* y PDF. También permiten el envío de correos cifrados y firmados [3].

Por otro lado, un administrador de sistemas informáticos puede solicitar su certificado digital para servidores web y servidores de correo. Estos certificados pueden ser utilizados para garantizar la autenticidad de un sitio web y de un servicio de correo. Además, permiten establecer un canal de comunicación seguro entre el servidor y el cliente.

4. Conclusión final y trabajos futuros

La implementación de una infraestructura de clave pública segura y confiable mediante el uso de herramientas *open source* y el desarrollo de módulos que permitan su adecuada implementación generan una nueva experiencia a nivel nacional donde se demuestra la viabilidad técnica para la universidad. Esta implementación permite que la universidad, como entidad referente de tecnología, pueda ofrecer los mecanismos necesarios para las operaciones electrónicas seguras dentro de la institución, y además deja las bases para ofrecer los mismos servicios a entidades externas.

Esta experiencia permite que la universidad sea una de las primeras instituciones en utilizar este tipo de mecanismo dentro de nuestro país, que marca una nueva tendencia en Paraguay, a pesar que ya se tienen marcadas experiencias en otros países como Brasil, Chile, Uruguay y España donde el uso es cotidiano y de validez jurídica.

Con la infraestructura en marcha, también se pueden hablar de otros aportes, tales como el desarrollo de una aplicación de autoridad de registro que sea capaz de integrarse con una aplicación de autoridad de certificación tan madura y difundida como es el caso de EJBCA. Cabe destacar el uso de herramientas libres en el desarrollo de la aplicación RA, lo cual permitirá seguir evolucionando y optimizando la aplicación tanto en funcionalidades como en seguridad.

Trabajos futuros pueden incluir implementaciones de servidores OCSP para la consulta online del estado de los certificados digitales, así como la implementación de un servidor *TimeStamp Authority* (TSA) para el montaje de una autoridad de sellado de tiempo. Esta autoridad de sellado de tiempo permitirá comprobar la existencia e integridad de documentos en un instante dado de tiempo, es decir, en una fecha y hora concreta [15]. También se podría implementar un repositorio LDAP de manera a contar con un servicio de directorio ordenado y distribuido de certificados digitales.

Otra área de trabajo a ser considerada es la implementación del API PKCS#11 para la emisión de *tokens hardware* en la aplicación RA. Esto permitirá que la aplicación RA, en conjunto con EJBCA, pueda emitir certificados digitales en dispositivos de *hardware* criptográficos tales como *smart-cards* y *tokens* USB. Esta implementación de mayor costo es relevante para proveer un mayor grado de seguridad en las operaciones criptográficas, especialmente cuando se tratan transacciones financieras y comerciales. En la aplicación RA también se pueden implementar otros módulos, tales como el modulo para la emisión de certificados a personas pertenecientes a empresas externas y el módulo para la emisión de certificados personales.

Referencias bibliográficas

[1]. Andrew S. Tanenbaum and David J. Wetherall. *Computer Networks*. Prentice Hall Press, Upper Saddle River, NJ, USA, 5th edition, 2010.

[2]. William Stallings. *Cryptography and Network Security: Principles and Practice*. Prentice Hall Press, Upper Saddle River, NJ, USA, 5th edition, 2010.

[3]. Carlos S. Cantero. *Infraestructura de Clave Pública para la Universidad Católica*. Tesis de grado de la Facultad de Ciencias y Tecnología de la carrera de Ingeniería Informática. Universidad Católica Nuestra Señora de la Asunción, 2013. http://www.cba.com.py/web/Resurces/PKI-SCantero.pdf. Último acceso: 24-07-2014.

[4]. Keyword search "*opensource pki*" in Google, July 2012.

[5]. Dogtag Certificate System. *Open Source* CA. http://pki.fedoraproject.org/. Último acceso: 25-04-2014.

[6]. Nicusor Vatra. *A PKI architecture using Open Source Software for E-goverment services in Romania*. Indian Journal of Computer Science and Engineering (IJCSE), 2, 2011.

[7]. Página oficial ejbca. http://www.ejbca.org/index.html. Último acceso: 25-04-2014.

[8]. M. Nystrom and B. Kaliski. *PKCS #10: Certification Request Syntax Specification Version 1.7*. RFC 2986 (Informational), Noviembre 2000.

[9]. D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 5280 (Proposed Standard), Mayo 2008.

[10]. Secretaría de Tecnologías de la Información y la Comunicación (SETICs). *Framework Tekoporu*. http://tekoporu.ticpy.org/. Último acceso: 11-11-2013.

[11]. Jboss community. http://www.jboss.org/index.html. Último acceso: 25-04-2014.

[12]. Documentación oficial PostgreSQL (versión 9.2). http://www.postgresql.org/files/documentation/pdf/9.2/postgresql-9.2-US.pdf. Último acceso: 25-04-2014.

[13]. Citrix xenserver. *Open-Source Server Virtualization*. http://www.citrix.com/products/xenserver/overview.html. Último acceso: 25-04-2014.

[14]. Centos. http://centos.org/index.html. Último acceso: 25-04-2014.

[15]. C. Adams, P. Cain, D. Pinkas, and R. Zuccherato. *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*. RFC 3161 (Proposed Standard), Agosto 2001.