Puntos clave para el desarrollo de una aplicación segura usando firma digital

Lic. Javier F. Diaz [jdiaz@info.unlp.edu.ar]
AC. Paula Venosa [pvenosa@info.unlp.edu.ar]
LINTI -Laboratorio de Investigación de Nuevas Tecnologías Informáticas -Facultad de Informática.
Universidad Nacional de La Plata. Buenos Aires. Argentina

Resumen:

El auge de la firma digital[1] para asegurar transacciones en Internet es un hecho que no se puede pasar por alto hoy en día. Si bien la firma digital de mensajes de correo electrónico y de transacciones WEB es muy importante, existen un sinnúmero de aplicaciones que se pueden beneficiar al introducir un esquema de seguridad que tenga como eje la firma digital (para garantizar, por ejemplo, autenticidad y no repudio).

El presente artículo describe los puntos más importantes para construir una aplicación de este tipo.

La propuesta se ilustra con un prototipo desarrollado y operativo en el LINTI[2], laboratorio de la Facultad de Informática de la Universidad Nacional de La Plata, llevado a cabo por Verónica Fredes y Paula Venosa.

Palabras claves

Internet – Seguridad – PKI - Firma Digital - Autoridad de certificación

[1] http://www.ncsc.dni.us/NCSC/TIS/ELECFILE/Digsig1.htm
[2] Laboratorio de Investigación en Nuevas Tecnologías Informáticas

Introducción

La proyección de Internet como medio para el comercio electrónico ha incentivado el crecimiento de la Red en los últimos años. Actualmente la conectividad a Internet es necesaria para todas las organizaciones, algunas de las cuales intentan resolver cómo conducir transacciones de misión crítica que se llevan a cabo sobre Internet así como sobre sus Intranets y Extranets.

Lógicamente, conforme más información hay disponible en Internet, más importancia cobra la protección de esa información y el control del acceso a la misma. Dentro de este panorama, nos enfrentamos a una creciente realidad, la necesidad de seguridad en los datos, los servicios, las transacciones, y las partes involucradas.

Sin ir más lejos, tomando como escenario el circuito administrativo que se lleva a cabo en nuestra Facultad durante la realización de cada trabajo de grado, al intentar automatizar ciertas instancias del mismo, descubrimos que existe la necesidad de cumplir con ciertas pautas relacionadas con la seguridad como por ejemplo garantizar que un alumno que envía un documento sea quien dice ser.

Por ello, aprovechando las ventajas ofrecidas por el servicio WEB, así como la posibilidad de efectuar transacciones seguras a través del uso de firma digital, trabajamos en el desarrollo de una aplicación cuyo objetivo será facilitar la interacción de quienes participan en este circuito, garantizando los requerimientos de seguridad que determinen que los trámites realizados a través del sistema son realmente válidos.

Por qué firma digital

Generalmente, en las transacciones basadas en documentación escrita existen mecanismos que garantizan: confidencialidad (asegura que el contenido del mensaje es privado[3]), autenticidad (garantiza que el mensaje proviene de la persona que efectivamente lo está enviando), integridad (asegura que el contenido del documento no ha sido modificado) y no repudio (no es posible que, finalizada la transacción, el emisor niegue haber enviado el documento o el receptor niegue haberlo recibido).

El objetivo de la seguridad es garantizar la privacidad de la información y la continuidad del servicio, tratando de minimizar la vulnerabilidad de los sistemas ó de la información contenida en ellos, así como tratando de proteger las redes privadas y sus recursos mientras que se mantienen los beneficios de la conexión a una red pública.

Dado que los servicios de Internet carecen de la implementación de estos conceptos de seguridad, los cuales son fundamentales para la validez legal de las transacciones de comercio electrónico, es necesario incorporar mecanismos que los garanticen. Las técnicas criptográficas, tales como encriptación y firma digital, constituyen piezas fundamentales para implementación de los mismos.

Objetivo

Nuestro objetivo se centró construir una aplicación que permita automatizar el circuito administrativo que se lleva a cabo para cada tesis, así como en diseñar e implementar una arquitectura que la soporte.

[3] Se entiende como privado un mensaje que no puede ser visto por ninguna persona excepto el destinatario.

Funcionalidad del sistema

Para comprender mejor cuales son los pasos que se intentarán automatizar al implementar el modelo que se describe en el presente artículo, presentamos a continuación algunos detalles[4].

Los usuarios que pueden interactuar con el sistema son: los alumnos en condiciones de realizar el trabajo de grado, los profesores admitidos para dirigir o evaluar trabajos de grado y los administrativos que cumplan alguna función relacionada con la gestión del sistema.

Cada usuario tendrá disponible distintas posibilidades dentro del sistema de acuerdo al rol que posea.

El alumno se reúne con su director y deciden el tema de la tesis además de como se conformará el grupo, recién entonces el director puede autorizar la emisión del certificado de los alumnos (previa verificación de los datos por parte del operador de la CA).

Cuando un alumno ingresa al sistema, según su estado se mostrará la pantalla correspondiente a la acción que podrá realizar en ese momento. Dicha acción puede consistir en realizar una entrega así como en confirmar o rechazar una entrega previamente realizada por alguno de sus compañeros. Cuando un alumno entrega un documento, los demás integrantes deben dar su consentimiento. Estos pasos deben cumplirse para cada entrega a efectuar.

La secuencia de informes a entregar es la siguiente:

- Propuesta
- Informe de avance
- Trabajo final

Recién después de que todos los miembros del grupo hayan consensuado respecto de la entrega, la misma podrá ser revisada por el director.

Si el director la aprueba, entonces esta entrega estará disponible para la revisión del jurado. En caso contrario, los alumnos se deberán comunicar personalmente con el director para llevar a cabo las correcciones pertinentes.

Los alumnos son notificados a través del correo electrónico de lo que deciden tanto su director como los jurados.

Se debe respetar el mismo circuito para las sucesivas entregas.

Requerimientos de seguridad en el sistema:

El análisis efectuado previamente al desarrollo dio como resultado la identificación de los siguientes problemas a resolver respecto a la seguridad del sistema:

En cuanto a las transacciones que se llevan a cabo:

1. El usuario debe tener la certeza de que realmente se está comunicando con el servidor Web de la Facultad.
2. Como el sistema debe informar ciertas acciones a través del correo electrónico, debe existir un mecanismo que garantice la autenticidad de los mensajes emitidos por el mismo.
3. Cuando un alumno envía un informe el sistema debe tener la seguridad de que realmente el alumno es quien dice ser.
4. El servidor almacenará un informe siempre y cuando el mismo no haya sido alterado durante su transmisión hasta el mismo.

[4] La reglamentación que regula este proceso se encuentra disponible en http://www.info.unlp.edu.ar/TrabajoDeGrado.htm

5. Que cuando un alumno revisa un informe escrito por un compañero el mismo no haya sido alterado durante su almacenamiento.

Conceptos relacionados

Antes de comenzar a describir nuestro proyecto, es necesario introducir ciertos conceptos que se harán presentes durante el desarrollo de este artículo.

Criptografía

Cuando un emisor y un receptor quieren intercambiar mensajes, es posible que un espía quiera intervenir de algún modo en la comunicación, a éste se lo conoce comúnmente como intruso. Un intruso puede ser pasivo, si sólo escucha la comunicación, o activo si trata de alterar los mensajes.

Es aquí donde aparece la criptografía con el objeto de proporcionar comunicaciones seguras sobre canales inseguros.

La criptografía[5], del griego "escritura oculta", ha evolucionado desde las técnicas de transformaciones y sustituciones de símbolos ya utilizadas en las antiguas civilizaciones, a los métodos basados en algoritmos matemáticos.

El objetivo de la criptografía[6] es proteger la información de forma tal que la misma tenga significado únicamente para el destinatario. Para ello la criptografía lleva a cabo el proceso de cifrado, transformando el texto plano[7] en texto cifrado.

Al proceso inverso, el cual transforma un mensaje cifrado en el texto plano correspondiente, se lo llama descifrado.

Idealmente los procesos de cifrado y descifrado no requieren demasiado esfuerzo, siempre que las claves involucradas sean conocidas.

En caso de que un intruso trate de descubrir el contenido del mensaje cifrado, el proceso de descifrado debería ser lo suficientemente costoso en lo que se refiere a tiempo y recursos invertidos en dicho intento, de modo tal de desanimar al espía.

Los algoritmos de cifrado tradicionales utilizan una única clave, la cual es compartida por el emisor y el receptor. El emisor usa dicha clave para cifrar el mensaje y el receptor realiza el proceso de descifrado utilizando también la clave en cuestión. Este mecanismo se conoce como *criptografía simétrica.*

En cambio, *la criptografía asimétrica o de clave pública* utiliza dos claves diferentes pero matemáticamente relacionadas. Dichas claves complementarias se denominan "clave privada" y "clave pública" respectivamente.

La idea general es proveer a cada usuario de un único par de claves (una pública y una privada) independientemente del número de usuarios con los que desee comunicarse. Estas claves tienen la propiedad que cada una de ellas invierte la acción de la otra pero, y aquí está el punto más relevante, a partir de una no se puede obtener la otra. Un usuario da a conocer su clave pública a otros usuarios y guarda su clave privada. Ambas claves pueden ser usadas para cifrar y descifrar datos.

---

[5] Criptografía: La criptografía es la ciencia que se encarga de mantener los mensajes en forma secreta

[6] Ref. http://www.rsa.com/

[7] Texto plano: La representación del mensaje original es conocida como texto plano.

Firma digital

La firma digital[8] está constituida por un conjunto o bloque de caracteres que viaja junto a un documento, archivo o mensaje y que puede acreditar cuál es el autor o emisor del mismo(lo que se denomina autenticación) y que nadie ha manipulado o modificado el mensaje en el transcurso de la comunicación (asegura la integridad del mensaje).

En términos más técnicos, una firma digital es una secuencia de bits que resultan del uso de una función de hash[9] o resumen que crea un digesto del mensaje transmitido en una comunicación electrónica. El digesto resultante es entonces encriptado usando un algoritmo de clave pública y la clave privada del emisor.

Un receptor que posee la clave pública del emisor del mensaje puede entonces saber si la secuencia de bits fue creada usando la clave privada del firmante y además comprobar si la comunicación ha sido alterada o no desde que la secuencia de bits fue generada.

Infraestructura PKI

El término PKI[10] se refiere, a veces, simplemente, a la jerarquía de confianza basada en el uso de certificados de clave pública. Una visión intermedia, en cambio, es que PKI incluye servicios y protocolos de manejo de clave pública a través de autoridades de certificación y autoridades de registración, pero no provee necesariamente operaciones criptográficas con las claves. La definición más completa entiende a PKI como los servicios de firma digital y criptografía provistos a las aplicaciones de usuario final.

Los sistemas basados en clave pública deben estar seguros de que cada vez que ellos confían en una clave pública, la clave privada es un secreto bien guardado[11] para todos salvo para el emisor de la información. Esta confianza se basa en el uso de certificados de clave pública los cuales son estructuras de datos que ligan valores de clave pública a los sujetos. La ligadura se realiza a través de una autoridad de certificación, la cual verifica la identidad del sujeto y firma digitalmente cada certificado.

Una **PKI** consta de cinco componentes fundamentales:

1. *Autoridades de certificación*, que emiten y revocan certificados.
2. *Autoridades de registración*, que atestiguan la asociación entre la clave pública y la entidad propietaria del certificado.
3. *Poseedores de los certificados emitidos* que pueden firmar documentos digitales.
4. *Entidad final*: usuarios del certificado PKI y/o usuarios del sistema que son sujeto de un certificado.
5. *Repositorios* que guardan y hacen disponibles tanto certificados como listas de revocación de certificados.

Si bien conceptualmente existe una definición y una descripción completa acerca PKI, no hay un framework simple en el cual sus componentes se combinen e interactúen en forma sencilla.

---

[8] Ref. http://www.baltimore.ie/clintonvisit98/techback.html

[9] Función de hash: Una función de hash o resúmen es un algoritmo que toma una entrada, generalmente de longitud indefinida, y generan un número pequeño de forma determinística (mensajes iguales siempre generan el mismo resultado).

[10] PKI: Public Key Infraestructure. Ref. http://www.entrust.com/downloads/docs/protocols_pki.htm

[11] Por secreto bien guardado se entiende que sólo su propietario tendrá acceso a la clave privada y se encargará de proteger y asegurar la misma.

Es por ello que en este punto, experimentamos varios problemas relacionados con la comunicación entre los componentes que probamos para implementar nuestro modelo, así como también existieron dificultades que tienen que ver con los formatos (ya sea de los certificados, como de los mensajes firmados y de las claves, entre otros) y los estándares definidos e implementados.

El Proyecto

Una arquitectura de estas características está compuesta principalmente por:

- Una Autoridad de Certificación: Entidad encargada de emitir los certificados para crear el contexto legal de la Firma Digital.
- Una interfaz de usuario: se basará en HTML o extensiones, de manera de poder usarla en un entorno como Internet.
- Un Servidor de WEB: el cual contendrá las páginas y scripts que implementarán la interfaz y relacionarán las partes del sistema. Dichos scripts serán los encargados de interactuar con la Base de Datos.
- Una Base de datos: Lugar donde se guardará y recuperará toda la información circunscripta como ser: registros de alumnos, estado de tesis, informes, etc.

Tareas realizadas

Para el desarrollo de este modelo tuvimos que estudiar, analizar y probar distintas herramientas. A grandes rasgos, los pasos realizados fueron:

- ✓ Una de las primeras tareas que realizamos fue el análisis de distintas Autoridades de Certificación para poder comprender la forma de trabajo de las mismas y el rol que cumplen dentro de la Infraestructura PKI. Los productos que instalamos y analizamos fueron: Microsoft Certificate Server, Netscape Enterprise Certificate Server y APuN-CA. Se probó la funcionalidad de cada una de ellas, prestando mayor atención en lo que se refiere a: requerimiento y emisión de certificados, políticas, revocación de certificados y procedimientos para la verificación de los datos.

  Llegado el momento de implementar nuestra propia infraestructura PKI la opción elegida fue APuN-CA[12], producto desarrollado por un grupo de trabajo de la Administración Pública Nacional Argentina, este producto permite construir una Infraestructura PKI que cuente con las características necesarias para proveer el Servicio de Certificación a una determinada comunidad de usuarios.

- ✓ Como lenguaje de programación del sistema elegimos JAVA aprovechando las ventajas provistas por el mismo como ser su robustez y versatilidad. Además de permitir a los programadores escribir programas en una plataforma y ejecutarlos en otra, así como crear programas que se ejecuten en el contexto de un navegador, desarrollar aplicaciones que corran del lado del servidor (para procesar un formulario HTML, por ejemplo), entre otras cosas.
- ✓ Como Servidor Web elegimos desde un principio Apache, principalmente por ser un software de libre distribución y por su popularidad y robustez. Además tuvimos que investigar los requerimientos adicionales de dicho servidor para contar con soporte para transacciones seguras y manejo de servlets. Se probaron distintas instalaciones

[12] http://www.pki.gov.ar

de los módulos requeridos debiendo compilarse el código fuente para que funcione con todos los módulos necesarios.
También experimentamos los problemas de compatibilidad existentes entre las distintas versiones del Servidor Web y los distintos módulos.

- ✓ Para contar con un repositorio de información instalamos una base de datos ORACLE.
- ✓ Para poder implementar todos los aspectos relacionados con criptografía buscamos librerías que nos proveyeran las operaciones necesarias. Estudiamos también la API de Java Security al igual que las librerías de IAIK[13].
- ✓ Para poder implementar el envío de mensajes firmados fue necesario comprender el manejo de tipos MIME.

Mecanismos usados para garantizar la seguridad:

SSL:

Para que exista un mínimo de protección se necesita contar con integridad de datos, confidencialidad de datos y autenticación de la entidad. SSL, al adicionar encriptación al protocolo http, brinda dichos servicios.

En nuestro modelo, se utiliza SSL cuando un cliente se comunica con el servidor WEB. Este último se autentica ante el cliente a través de un certificado emitido por la CA "Autoridad certificante de la Facultad de Informática" que forma parte de nuestra arquitectura. Al mismo tiempo el Servidor requiere que el cliente se autentique utilizado un certificado emitido también por dicha autoridad de certificación.

Los pasos llevados a cabo durante la conexión no son totalmente transparentes al usuario, el cual debe seleccionar el certificado adecuado entre todos aquellos que tiene instalados en su Navegador.

Luego de haber instalado Apache como servidor WEB, fue necesario instalar un módulo adicional para contar con el soporte para SSL así como también debimos configurar el mismo incluyendo las directivas adecuadas para que se produzca la autenticación en ambos extremos.

Mecanismos de firma:

Con el fin de asegurar la validez de las operaciones, las mismas son dotadas del servicio de firma digital. Ello ocurre cada vez que se envía un informe o un formulario (para realizar una confirmación de un documento, para ingresar un nuevo grupo de tesistas, etc) a través de la interfaz web.

El proceso de firma se realiza a través de un applet (tanto en el caso en el cual el objeto a firmar es un documento como en el caso de que el objeto a firmar sea un formulario) con lo cual se logra que la clave privada no sea transmitida, dado que el proceso se ejecuta en el cliente.

Una vez realizada la firma, la misma es enviada al servidor. El servidor verifica la firma recibida, lo cual se lleva a cabo en un servlet, el cual cuenta con los elementos necesarios para llevar a cabo la verificación: el objeto que fue firmado, la firma y la clave pública del firmante (la cual es conocida al existir una conexión SSL, en el ambiente del servidor puede accederse a una variable de ambiente que contiene toda la información incluida en el certificado del cliente actual).

Además, cada vez que una transacción se lleva a cabo, el sistema dispara mensajes de correo electrónico para informar a las partes que se vean afectadas por dicha transacción, el estado de la

[13] Ref. http://www.iaik.at/

tesis involucrada en la misma. Así como el envío de formularios y de documentos, el envío de mensajes de correo también se realiza usando firma digital, en este caso el sistema es el que firma los mensajes.

IAIK-JCE

Para el manejo de operaciones relacionadas con la seguridad fue necesario comprender el modelo de seguridad de JAVA así como incorporar librerías que nos provean los métodos necesarios para poder implementar las funciones de firma y verificación de la firma.

IAIK-JCE o "IAIK Java Cryptography Extension" es un conjunto de APIs e implementaciones de funciones criptográficas, incluyendo cifrado simétrico, asimétrico, cifrado de flujo y de bloque.

Provee una re-implementación de las Extensiones Criptográficas de Java.La arquitectura de IAIK-JCE sigue los mismos principios de diseño utilizado por JCA.

IAIK-JCE cuenta con su propio proveedor de seguridad, el cual ofrece una gran variedad de servicios criptográficos y algoritmos. La clase principal del paquete IAIK security provider es la clase IAIK del paquete iaik.security.provider. Este extiende la clase java.security.Provider para incorporar las implementaciones específicas de seguridad que facilita el proveedor IAIK.

La API está equipada con una librería para manejo de ASN.1. Soporta además los siguientes estándars PKCS: PKCS#1, PKCS#5, PKCS#7, PKCS#8, PKCS#10, PKCS#12.

También puede ser usada para manejar certificados X509. La arquitectura de certificados X509 y de CRLs implementada por IAIK-JCE se adapta fácilmente a la estructura de certificados impuesta por java.security. IAIK-JCE soporta todos las extensiones estándares existentes para certificados X509 V3 así como las extensiones más importantes de las CRLs[14] X509 V3 y de los certificados Netscape.

IAIK-JCE Applet Edition

Es la versión para applets de la librería IAIK-JCE, la cual es absolutamente idéntica a la librería estándar IAIK-JCE salvo en los cambios (re-implementación de algunas clases) y características adicionales que incorpora por ser necesarios para que las clases funcionen correctamente en todos los navegadores compatibles con JDK 1.1.

IAIK-S/MIME:

IAIK es una implementación de la versión 2 del protocolo S/MIME 100% desarrollada en JAVA.

Para cumplir con su propósito de extender el estándar MIME adicionando al mismo los servicios de seguridad de autenticación, integridad del mensaje, no repudio del origen y privacidad y seguridad de datos, IAIK-S/MIME opera junto a IAIK-JCE.

IAIK-S/MIME se basa fundamentalmente en el paquete iaik.pkcs.pkcs7 de IAIK-JCE. De esta forma, IAIK S/MIME aprovecha las ventajas de la implementación de stream de la librería PKCS7 posibilitando manejar grandes cantidades de datos sin que ello traiga aparejado problemas de memoria o de performance. Por esta razón, la clase SMimeSigned del paquete

[14] CRL(Certificate Revocation List): Lista de revocación de certificados

iaik.security.smime extiende la clase SignedDataStream del paquete iaik.pkcs.pkcs7, y la clase SmimeEncrypted extiende la clase EnvelopedDataStream.

IAIK-S/MIME puede ser incorporada fácilmente a la API javax.mail de SUN. La misma soporta los siguientes tipos para los mensajes de correo electrónico:

- ✓ signed (multipart/signed; application/pkcs7-mime).
- ✓ encrypted (application/pkcs7-mime).
- ✓ signed and encrypted.
- ✓ certs only.
- ✓ certificate requests (application/pkcs10).

Solución desarrollada

Luego del período de investigación de los temas relacionados al proyecto y de llevar a cabo las pruebas, el diseño y la implementación de varias partes de este sistema, estamos en condiciones de mostrar las soluciones propuestas a los requerimientos de seguridad anteriormente mencionados.

1. El sistema fue montado sobre un servidor WEB seguro, el cual utiliza SSL, con lo cual el usuario tiene la garantía de que se está comunicando con el Servidor con el que desea hacerlo, ya que es el único capaz de desencriptar la información encriptada con su clave pública, por ser el poseedor de la clave privada correspondiente.
2. Para garantizar la autenticidad de los mensajes de correo electrónico emitidos por el sistema, al implementar el envío de los mismos, éstos se firman digitalmente, de modo tal que el usuario que los reciba pueda corroborar el origen de los mismos.
3. Cuando un alumno envía un informe, un applet se encarga previamente de firmar digitalmente el documento en cuestión, usando para ello la clave privada que reside en el cliente.
4. Cuando el servidor recibe un informe lleva a cabo el correspondiente proceso de verificación a partir de la firma, el objeto firmado y la clave pública de quien envía el informe.
5. Cuando un alumno accede al informe enviado por un compañero suyo, tiene la plena garantía de que el mismo no ha sido modificado porque el Servidor realiza el proceso de verificación antes de presentarle el informe. La realización de la verificación es factible porque el Servidor almacena tanto el informe como la firma del mismo y el certificado de clave pública de quien lo ha firmado.

Dado que se usarán mecanismos de firma digital para garantizar la validez de las operaciones debe cumplirse, además de los requerimientos de seguridad anteriormente descriptos para las transacciones, que la clave privada de un usuario no será transmitida nunca y que el certificado de clave pública de cada parte es válido, esto último se garantiza con la existencia de la autoridad de certificación como parte de una infraestructura de clave pública mientras que, al efectuarse la operación de firmado del lado del cliente aseguramos que la clave privada será sólo conocida por el cliente.

Conclusiones

El aporte más importante de este trabajo es la experiencia obtenida a partir del desarrollo de este tipo de aplicación, la cual constituye una base para futuros desarrollos.

Existen otros puntos que no queremos dejar de mencionar. Entre ellos el hecho de que la PKI desarrollada es monolítica. En caso de que se quiera hacer un uso extendido de la misma, entendiéndose por ello una arquitectura distribuida, deberían tenerse en cuenta otros factores como ser la construcción de la estructura de CAs y la creación de políticas para emitir certificados afines.

Además, este sistema, el cual se encuentra actualmente en una etapa de testing, no posee ningún requerimiento de hardware específico.

Referencias:

"Electronic Commerce - On-line Odering Digital Money"
Segunda edición
Pete Loshin/ Paul Murphy
Charles River Media, INC.

"Digital Certificates – Applied Internet Security"
Jalal Feghhi - Jalil Feghhi - Peter Williams"
Addison Wesley

"Secure Electronic Commerce – Building the infraestructore for Digital Signatures and Encryption"
Warwick Ford – Michael S. Baum
Prentice Hall PTR

http://www.modernizacion.com.cl/utic/firma/index.htm
http://www.w3.org/Security/Faq/www-security-faq.html