PROTOCOLO DE SEGURIDAD SSL

Resumen / *Abstract*

El creciente uso de Internet, ha dirigido la atención hacia un problema crucial: la privacidad. Para que tenga lugar una negociación en Internet, se precisa, en la mayoría de los casos, que cada entidad participante pueda contar con una manera eficaz de verificar la identidad de las otras y establecer un nivel de confianza. Es necesario, por tanto, crear un entorno que garantice la autenticidad y seguridad de las transacciones que tienen lugar en este proceso. SSL (Secure Sockets Layer) es el protocolo de seguridad más estandarizado que, haciendo uso de un conjunto de técnicas criptográficas, asegura confidencialidad e integridad de la información.

The increasing use of Internet has drawn the attention to a crucial problem: privacy. In order to carry out negotiations in Internet, it's needed, most of the time that each party involved counts on the possibility of efficiently verifying other's identity and reach a high level of confidence. That is why, it is necessary to create an environment which guarantees the authenticity and safety of transactions involved in this process. SSL (Secure Sockets Layer) is the most standardized security protocol that, by means of a set of cryptographic techniques, it assures confidentiality and integrity of information.

Palabras clave *Key words*

Secure sockets layer, SSL, seguridad, encriptación, certificado digital, llave pública, llave privada, firma digital, HTTP, SelfSSL, OpenSSL

Secure sockets layers, SSL, security, encryption, digital certificate, public key, private key, digital signature, HTTP, SelfSSL, OpenSSL

INTRODUCCIÓN

Es un hecho que Internet constituye un canal de comunicaciones inseguro, debido a que la información que circula a través de la red es fácilmente accesible en cualquier punto intermedio por un posible atacante. Los datos transmitidos entre dos nodos de Internet, se segmentan en pequeños paquetes que son encaminados a través de un número variable de nodos intermedios hasta que alcanzan su destino. En cualquiera de ellos es posible leer el contenido de los paquetes, destruirlo e incluso modificarlo, posibilitando todo tipo de ataques contra la confidencialidad y la integridad de los datos. En el caso de que se necesite enviar datos confidenciales, la solución más comúnmente adoptada se basa en la utilización del protocolo SSL.

SSL Y LAS TÉCNICAS DE LA CRIPTOGRAFÍA

SSL (Secure Sockets Layer) es un protocolo de propósito general para establecer comunicaciones seguras, propuesto en 1994 por Netscape Communications Corporation junto con su primera versión del Navigator. Sin embargo, no fue hasta su tercera versión, conocida como SSL v3.0 que alcanzó su madurez, superando los problemas de seguridad y limitaciones de sus predecesores. No es exclusivo del comercio electrónico sino que sirve para cualquier comunicación vía Internet y, por lo tanto,

Sandra Ortega Martorell, *Ingeniera Informática, Instructora Centro de Estudios de Ingeniería de Sistemas (CEIS), Instituto Superior Politécnico José Antonio Echeverría, Ciudad de La Habana, Cuba*
e-mail:sandra@ceis.cujae.edu.cu

Liusbetty Canino Gutiérrez, *Ingeniera Informática, CEIS, Instituto Superior Politécnico José Antonio Echeverría, Ciudad de La Habana, Cuba*
e-mail:lcanino@ceis.cujae.edu.cu

Recibido:mayo del 2006
Aprobado:junio del 2006

también para transacciones económicas. SSL está incorporado a muchos navegadores web además del Navigator de Netscape, y el Internet Explorer de Microsoft. Hoy constituye la solución de seguridad implantada en la mayoría de los servidores web que ofrecen servicios de comercio electrónico.

SSL está basado en la aplicación conjunta de criptografía simétrica (de llave secreta), criptografía asimétrica (de llave pública), certificados digitales y firmas digitales para conseguir un canal o medio seguro de comunicación a través de Internet.[1]

De los sistemas criptográficos simétricos, motor principal de la encriptación de datos transferidos en la comunicación, se aprovecha la rapidez de operación. Estos sistemas adicionan códigos de autenticación de mensajes (MAC por sus siglas en inglés) para garantizar la integridad de los datos. Por su parte, los sistemas asimétricos se usan para el intercambio seguro de las claves simétricas, consiguiendo con ello resolver el problema de la confidencialidad en la transmisión de datos. La identidad de un servidor web seguro (y a veces también del usuario cliente) se consigue mediante el certificado digital correspondiente, del que se comprueba su validez antes de iniciar el intercambio de datos sensibles, mientras que de la seguridad de la integridad de los datos intercambiados se encarga la firma digital mediante funciones *hash* y la comprobación de resúmenes de todos los datos enviados y recibidos.

La criptografía de llave secreta resulta útil en muchos casos, aunque tiene limitaciones significativas. Todas las partes deben conocerse y confiar totalmente la una en la otra, pues se utiliza una única clave tanto para encriptar como para desencriptar la comunicación, por tanto cada parte debe poseer una copia de la llave. Por sí solo, este tipo de encriptación no es suficiente para desarrollar a plenitud, por ejemplo, el potencial del comercio electrónico, el cual vincula a un número ilimitado de compradores y vendedores de todas partes del mundo. De un lado, resulta poco práctico que una gran corporación intercambie claves con miles o incluso millones de clientes o, peor todavía, con posibles clientes con los que nunca ha tratado.[2]

La solución a la seguridad en toda red abierta es la criptografía de llave pública, una forma de codificación más novedosa y sofisticada. En este tipo de enfoque, cada participante crea dos claves únicas. Se dispone de una clave pública, que se publica en un tipo de directorio al que el público en general tiene acceso y se dispone, además, de una clave privada, que se mantiene en secreto. Las dos claves funcionan conjuntamente como un curioso dúo. Cualquier tipo de datos o información que una de las claves **cierre**, solo podrá abrirse con la otra. De esta forma, en una comunicación entre dos partes cualesquiera, una busca la clave pública de la otra y la utiliza para encriptar el texto. El receptor del mismo utiliza su clave privada para revertir la encriptación del mensaje en la pantalla de su computadora y aparece el mensaje en forma de texto normal y corriente. Si un extraño interceptara este mensaje, no podría descifrarlo porque no tendría la clave privada de la otra parte.[2]

Los certificados digitales son una forma de agregar un tercer **árbitro** a la cadena de confianza de la comunicación por SSL. Lo que un certificado digital hace es agregar el **endoso** de un tercero que garantiza la integridad, y existencia de la organización que envía los datos. Esto significa que una autoridad certificadora avala que la empresa que es dueña del sitio web, por ejemplo, realmente existe.[3]

Un certificado digital es un documento digital publicado por una autoridad de certificación (CA por sus siglas en inglés). Un certificado digital incluye el nombre del sujeto (la compañía o el individuo que está siendo certificado), la clave pública del sujeto, un número de serie, una fecha de expiración, la firma de la autoridad de certificación, y cualquier otra información relevante. Una CA es una institución financiera u otra parte confiable, como VeriSign. La autoridad de certificación es responsable de la autenticación, de manera que debe chequear cuidadosamente la información antes de publicar un certificado digital. Los certificados digitales están disponibles públicamente y son contenidos por las autoridades de certificación en repositorios de certificados.

La CA firma el certificado ya sea encriptando la clave pública o un valor de *hash* de la clave pública, utilizando su propia clave privada. La CA tiene que verificar cada clave pública individual. De esa manera, los usuarios deben confiar en la clave pública de la CA. Usualmente, cada CA es parte de una jerarquía de autoridades certificadoras, dicha jerarquía es a su vez una cadena de certificados, empezando por la autoridad raíz de certificación, la cual es la Internet Policy Registration Authority (IPRA). La IPRA firma certificados utilizando la clave raíz. La raíz solo firma certificados para autoridades de creación de políticas (Policy Creation Authorities), las cuales son organizaciones que establecen políticas para obtener certificados digitales. En orden, las autoridades de creación de políticas, firman los certificados digitales para las CA, y las CA firman los certificados digitales para individuos y organizaciones.

FUNCIONAMIENTO DE SSL

En el modelo de referencia TCP/IP, SSL se introduce como una especie de nivel o capa adicional, situada entre la capa de **aplicación** y la capa de **transporte** (figura 1). Lo anterior hace que sea independiente de la aplicación que lo utilice, es decir, que no solo puede ser utilizado para encriptar la comunicación entre un navegador y un servidor Web, sino también en cualquier aplicación como IMAP, FTP, Telnet, etc. También puede aplicar algoritmos de compresión a los datos a enviar y fragmentar los bloques de tamaño mayor a 214 bytes, volviendo a reensamblarlos en el receptor.[1] Además, SSL establece una comunicación segura a nivel de **socket** (nombre de máquina más puerto), de forma transparente al usuario y a las aplicaciones que lo usan.

SSL es muy flexible con respecto a escoger el algoritmo de encriptación simétrico, la función de verificación de mensaje y el método de autenticación. La combinación de los elementos anteriores es conocida como suite de cifrado (Cipher Suite). Para

la encriptación simétrica SSL puede usar los algoritmos DES (Data Encryption Standard), Triple DES, RC2, RC4, Fortezza e IDEA; para la verificación de mensajes puede usar MD5 (Message Digest Algorithm 5) o SHA-1 (Secure Hash Algorithm) como algoritmos de *hashing* y para la autenticación puede usar algoritmos RSA (Rivest, Shamir, Adelman) u operar en modo anónimo en donde se usa el intercambio de llaves de Diffie-Hellman.[4] Los algoritmos, longitudes de clave y funciones *hash* usados en SSL dependen del nivel de seguridad que se busque o se permita.

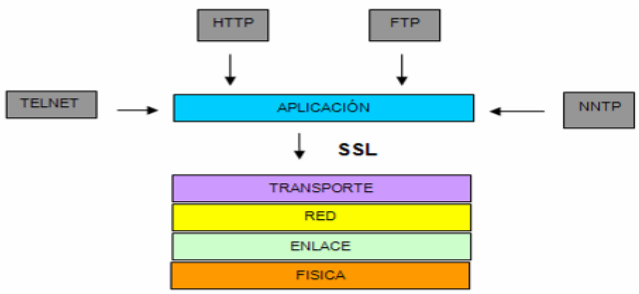

Fig. 1 *SSL entre la capa de Aplicación y la capa de Transporte.*

COMUNICACIÓN SSL

Para establecer una comunicación segura utilizando SSL se tienen que seguir una serie de pasos. Primero se debe hacer una solicitud de seguridad. Después de haberla hecho, se deben establecer los parámetros que se utilizarán para SSL. Esta parte se conoce como SSL *handshake*. Una vez que se haya establecido una comunicación segura, se deben hacer verificaciones periódicas para garantizar que la comunicación sigua siendo segura a medida que se transmitan datos. Luego que la transacción ha sido completada, se termina SSL.[5]

Solicitud de SSL

Antes de que se establezca SSL, se debe hacer una solicitud. Típicamente esto implica un cliente haciendo una solicitud de un URL a un servidor que soporte SSL. SSL acepta solicitudes por un puerto diferente al utilizado normalmente para ese servicio.[5]

Una vez se ha hecho la solicitud, el cliente y el servidor empiezan a negociar la conexión SSL, es decir, hacen el SSL *handshake*.[5]

SSL Handshake

Durante *el hanshake* se cumplen varios propósitos. Se hace autenticación del servidor y opcionalmente del cliente, se determinan qué algoritmos de criptografía serán utilizados y se genera una llave secreta para ser utilizada durante el intercambio de mensajes subsiguientes durante la comunicación SSL. Los pasos que se siguen son los siguientes:[5]

Client Hello: El saludo de cliente tiene por objetivo informar al servidor qué algoritmos de criptografía puede utilizar y solicita una verificación de la identidad del servidor. El cliente envía el conjunto de algoritmos de criptografía y compresión que soporta, y un número aleatorio. El propósito del número aleatorio es para que en caso de que el servidor no posea un certificado para comprobar su identidad, aún se pueda establecer una comunicación segura utilizando un conjunto distinto de algoritmos. Dentro de los protocolos de criptografía hay un protocolo de intercambio de llave que define cómo cliente y servidor van a intercambiar la información, los algoritmos de llave secreta que definen que métodos pueden utilizar y un algoritmo de *hash* de una sola vía. Hasta este punto no se ha intercambiado información secreta, solo una lista de opciones.

Server Hello: El servidor responde enviando su identificador digital el cual incluye su llave pública, el conjunto de algoritmos criptográficos y de compresión y otro número aleatorio. La decisión de qué algoritmos serán utilizados está basada en el más fuerte que tanto cliente como servidor soporten. En algunas situaciones el servidor también puede solicitar al cliente que se identifique solicitando un identificador digital.

Aprobación del cliente: El cliente verifica la validez del identificador digital o certificado enviado por el servidor. Esto se lleva a cabo desencriptando el certificado utilizando la llave pública del emisor y determinando si este proviene de una entidad certificadora de confianza. Después se hace una serie de verificaciones sobre el certificado, tales como fecha, URL del servidor, etc. Una vez se ha verificado la autenticidad de la identidad del servidor. El cliente genera una llave aleatoria y la encripta utilizando la llave pública del servidor y el algoritmo criptográfico y de compresión seleccionado anteriormente. Esta llave se le envía al servidor y en caso de que el *handshake* tenga éxito será utilizada en el envío de futuros mensajes durante la sesión.

Verificación: En este punto ambas partes conocen la llave secreta, el cliente porque la generó y el servidor porque le fue enviada utilizando su llave pública, siendo la única forma posible de desencriptarla utilizando la llave privada del servidor. Se hace una última verificación para comprobar si la información transmitida hasta el momento no ha sido alterada. Ambas partes se envían una copia de las anteriores transacciones encriptada con la llave secreta. Si ambas partes confirman la validez de las transacciones, el *handshake* se completa, de otra forma se reinicia el proceso.

Ahora ambas partes están listas para intercambiar información de manera segura utilizando la llave secreta acordada y los algoritmos criptográficos y de compresión. El *handshake* se realiza solo una vez y se utiliza una llave secreta por sesión.

Intercambio de datos

Ahora que se ha establecido un canal de transmisión seguro SSL, es posible el intercambio de datos.

Terminación de una sesión SSL

Cuando el cliente deja una sesión SSL, generalmente la aplicación presenta un mensaje advirtiendo que la comunicación no es segura y confirma que el cliente efectivamente desea abandonar la sesión SSL.[5]

Las especificaciones técnicas de la implementación de SSL se encuentran en el sitio oficial de Netscape.[6]

VERSIONES DEL PROTOCOLO SSL

En 1994, Netscape Communications creó SSL v.2, la cual hacía posible mantener la confidencialidad en los números de las tarjetas de crédito y además autenticar al servidor web con el uso de encriptación y certificados digitales. En 1995, Netscape fortalece los algoritmos criptográficos y soluciona muchos de los problemas de seguridad existentes en SSL v.2 con la nueva versión SSL v.3, la cual soporta más algoritmos de seguridad que SSL v.2. [7] Por otra parte Internet Engineering Task Force (IETF) adoptó SSL para la creación de su protocolo Transport Layer Security (TLS) y WAP Forum adaptó este último para crear el protocolo inalámbrico equivalente, Wireless Transport Layer Security (WTLS).[8] Conceptualmente, SSL, TLS y WTLS proveen el mismo servicio de seguridad: un canal seguro entre dos entidades, un cliente y un servidor.

Los navegadores más populares en la actualidad implementan SSL/TLS por defecto. Netscape Communicator (4.7) soporta solamente SSL y no TLS, mientras Netscape 6, MS Internet Explorer y Opera ofrecen soporte para ambos.[8] Es importante señalar en este punto, además, que los navegadores mencionados anteriormente soportan todos los algoritmos simétricos RC2, RC4, DES y Triple DES y las funciones *hash* MD5 y SHA-1. Todos ellos brindan soporte total para RSA mientras que el establecimiento de llaves Diffie-Hellman todavía no se vislumbra.

Diferencias entre SSL v.2 y SSL v.3

La primera versión pública de SSL, versión 2, tenía un conjunto de desperfectos de seguridad que fueron corregidos en SSL v.3. Los navegadores en la actualidad todavía soportan SSL v.2 y en muchos otros sistemas todavía está en uso. He aquí un resumen de los principales problemas:[8]

• Las mismas llaves criptográficas son usadas para la autenticación de mensajes y para encriptar, lo cual significa que los MACs de los mensajes están innecesariamente debilitados (debido a las restricciones de exportación de los Estados Unidos, la longitud de la llave simétrica que puede usar Netscape e Internet Explorer fue limitada a 40 bits. Si la llave de encriptación es usada también para la autenticación de mensajes, la seguridad de los MACs es además afectada).

• SSL v.2 no posee ninguna protección para la negociación (*handshake*) que tienen lugar entre cliente y servidor para establecer el intercambio de información, por tanto un ataque persona-en-el-medio no puede ser detectado.

• Finalmente, SSL v.2 simplemente utiliza el cerrado de conexión de TCP para indicar el fin del envío de datos, por tanto un ataque puede sencillamente falsificar los TCP FINs y el receptor no puede decir que no es un fin de envío de datos legítimo (SSL v.3 soluciona este problema implementando una alerta de clausura explícita).

Anterior a la propuesta de SSL v.3, estos problemas de seguridad fueron resueltos por Microsoft en su protocolo Private Communications Technology (PCT), el cual es muy similar a SSL/TLS y es aún soportado por los navegadores y servidores de Microsoft, pero SSL/TLS se ha convertido en un estándar y PCT no se ha podido imponer.

En adición al soporte para nuevos algoritmos de seguridad que ofrece SSL v.3, incorpora soporte para la carga de certificados en cadena*. Esta característica permite al servidor pasar un certificado del servidor junto con los certificados de otros emisores al navegador. Según la especificación del protocolo,[9] los objetivos de SSL v.3, en orden de prioridad, son los siguientes:

• **Seguridad criptográfica:** SSL debe ser usado para establecer una conexión segura entre dos partes.

• **Interoperabilidad:** Programadores independientes deben ser capaces de desarrollar aplicaciones utilizando SSL v.3 que puedan ser capaces de intercambiar satisfactoriamente parámetros criptográficos sin el conocimiento del código de otro.

• **Extensibilidad:** SSL v.3 pretende proveer un *framework* en el que nuevas llaves públicas y métodos de encriptación puedan ser incorporados si es necesario. Esto además conlleva dos objetivos: prevenir la necesidad de crear un nuevo protocolo (con el riesgo de introducir nuevas debilidades) y evitar la necesidad de implementar una nueva librería de seguridad completa.

• **Eficiencia relativa:** Las operaciones criptográficas tienden a hacer un uso intensivo del CPU, particularmente las operaciones de llave pública. Por esta razón, el protocolo SSL ha incorporado un esquema opcional de sesión oculta para reducir el número de conexiones que necesitan ser establecidas. Además, se han tomado precauciones para reducir la actividad en la red.

Diferencias entre SSL v.3 y TLS

El grupo de trabajo IETF adoptó el protocolo SSL v.3 y llevó a cabo pequeñas modificaciones para incrementar la seguridad, lo cual trajo como resultado el protocolo TLS. Algunas de estas modificaciones aparecen a continuación:[8]

• Las llaves criptográficas son ampliadas a partir de la mejora en el secreto intercambiado.

• La construcción del MAC fue modificado ligeramente apareciendo un HMAC.

• Las implementaciones requirieron incluir soporte para el protocolo de intercambio de llaves Diffie-Hellman, el estándar de firma digital y para el algoritmo Triple-DES.

• Adiciona estandarización en el orden de los mensajes, más mensajes de alerta y más bloques de relleno a los bloques codificados.

Diferencias entre TLS y WTLS

El WAP Forum ha adaptado TLS para introducirlo en el entorno inalámbrico en dispositivos pequeños, los cuales tienen limitaciones en el ancho de banda, memoria y procesamiento. Las nuevas características son:[8]

• WTLS incluye el uso de criptografía de curva elíptica por defecto.

*En una cadena de certificados, cada entidad certifica a la que le antecede. Esta cadena es importante en los casos en que la primera línea de autoridades que emiten los certificados no son tan conocidas o confiables como las últimas.

• WTLS trabaja encima del datagrama en lugar de la capa de comunicación basada en conexión (comparado con UDP vs TCP en Internet).

• WTLS define su propio formato para los certificados optimizados por el tamaño pero también soporta el certificado común X.509v3.

INSTALAR SSL EN IIS

La instalación de SSL depende del servidor web a emplear, a continuación se muestra cómo se pudiera instalar para Internet Information Server (IIS).

Instalar SSL en IIS para usarlo como prueba

Para instalar SSL en un servidor Windows se puede utilizar SelfSSL del paquete IIS Resource Kit (http://www.microsoft.com/downloads/details.aspx?FamilyID=56fc92ee-a71a-4c73-b628-ade629c89499&displaylang=en).

SelfSSL versión 1.0 es una herramienta que se utiliza para generar e instalar certificados SSL con firmas propias para IIS 6.0. Aunque fue diseñado para IIS 6.0, también funciona correctamente para IIS 5.1. Como SelfSSL genera certificados con firmas propias que no son de una fuente confiable conocida, la herramienta es útil para dos propósitos fundamentales:

• Cuando se quiere crear un canal seguro privado entre el servidor y un limitado y conocido grupo de usuarios, por ejemplo, en un ambiente de desarrollo de software. Para establecer este canal, se debe enviar una copia del certificado a los clientes que usarán el sitio Web, para que añadan el certificado a la lista de certificados confiables.

• Cuando se necesite solucionar problemas de certificados de terceras partes. Si se genera satisfactoriamente un certificado con SelfSSL y se instala en el IIS, entonces se sabrá que el IIS está funcionando correctamente. En ese caso se debe contactar a la entidad emisora (tercera parte) del certificado.

A continuación se explica cómo instalar SelfSSL versión 1.0 de IIS Resource Kit e instalar el certificado en IIS 6.0. Al ejecutar iis60rkt.exe, saldrá una pantalla de inicio de instalación, se deberá hacer clic sobre el botón **siguiente**, luego aparece la licencia de uso y se debe indicar que se está de acuerdo. Al hacer clic nuevamente en el botón **siguiente**, aparece una pantalla para entrar información del usuario. Al llenar los datos se avanza a la siguiente pantalla donde se selecciona el tipo de instalación (seleccionar **custom**) y se vuelve a hacer clic en **siguiente**. En este punto es donde se indica el camino donde se instalará, este se podrá cambiar si se desea. Al hacer clic en el botón **siguiente** se mostrará un cuadro donde se podrá indicar que se desea instalar "SelfSSL". El resto de las opciones se pueden desmarcar. AL hacer clic en **siguiente,** se mostrará un resumen. Hacer clic en **siguiente** nuevamente para comenzar la instalación. Cuando esta termine, hacer clic en el botón **finalizar**.

Ahora se podrá crear un certificado. Hacer clic en **Inicio** -> **Todos los Programas** -> **IIS Resources** -> **SelfSSL** -> **SelfSSL**. Se mostrará una ventana de comandos de Windows. Leer las indicaciones que aparecen en el cuadro para mayor información. Escribir "**selfssl /T**", sin las comillas. Escribir "**y**" cuando se pregunte si se desea remplazar la configuración de SSL.

En el explorador de Internet abrir **https://localhost**, e indicar que se desea ver un sitio seguro. SelfSSL crea un certificado para la realización de pruebas. La conexión será segura, pero cada vez que se cargue la página preguntará si desea abrirla, dado que no reconoce la entidad certificadora.

Solicitar un certificado digital con IIS

Abrir la consola de configuración de IIS, y abrir las propiedades del sitio Web predeterminado haciendo clic derecho en "Default Web Site". En la pestaña de seguridad, hacer clic en "**Server Certificate**". IIS hace casi de modo automático todas las labores necesarias para solicitar e instalar un certificado digital. Esto consiste en generar una clave privada, una clave pública y enviar a la entidad certificadora la clave pública para que la valide, firme y genere el certificado. Aparecerá una ventana para la creación del certificado en el servidor web. Más adelante, aparecerá un cuadro que permitirá indicar que se trata de la preparación de una solicitud de certificado que será enviada más tarde. Luego se deberá llenar los datos necesarios para generar el certificado, tales como: nombre para reconocerlo y longitud de la clave, datos de la organización, nombre de la máquina. Este punto es importante porque un certificado solo vale para un nombre de dominio válido. Si el nombre cambia, se deberá obtener un nuevo certificado. Se introduce además la información geográfica, y a continuación aparecerá un cuadro donde se indica en qué lugar se almacenará el fichero que contiene la información de la solicitud. Terminado este paso se muestra un cuadro, en el que se presenta un resumen de los datos para que sean comprobados. Al hacer clic en **siguiente** se podrá finalizar la creación del certificado.

El fichero generado se envía a una entidad certificadora, la cual enviará de regreso un fichero que contiene un nuevo certificado. Reiniciar este procedimiento para añadir el nuevo certificado al servidor. Aparecerá un cuadro como el que se muestra en la figura 2, a partir de aquí se desencadenará el proceso de añadir el nuevo certificado:

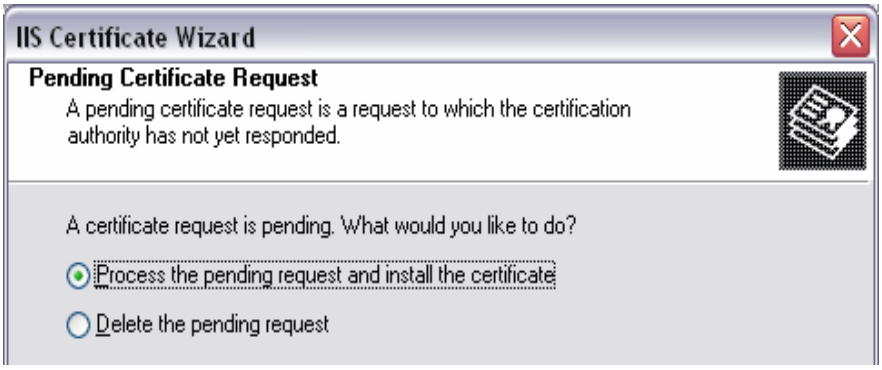

Fig. 2 *Pantalla para añadir un nuevo certificado.*

Otra manera de solicitar un certificado es ir directamente al sitio *online* de una entidad certificadora.

Crear una entidad certificadora

Como se indicaba anteriormente, para crear un certificado se deberá contactar a una empresa que los emita o, si se quiere solo para probar o para usarlo en la intranet, se puede crear una entidad certificadora propia y generar los certificados. Esto se puede lograr con programas como OpenSSL, que puede ser descargado de www.openssl.com. Para el presente trabajo se descargó de Internet una versión compilada llamada Win32 OpenSSL, que es una versión de OpenSSL para Windows. Una vez descargado el software, se procede a su instalación en la máquina.

El primer paso será generar una clave privada de la entidad certificadora. Para ello se deberá abrir la ventana de comandos de Windows (Inicio->Run: cmd.exe), y desde el directorio donde se encuentra instalado OpenSSL, ejecutar openssl.exe de la siguiente manera (figura 3):

openssl genrsa -des3 -out cakey.pem 2048

Se solicitará que entre una contraseña, y luego que la vuelva a entrar para verificarla. Se generará la clave en un fichero denominado cakey.pem en la carpeta donde está instalado OpenSSL. Luego se crea un certificado digital, para ello se escribe la siguiente línea:

openssl req -new -x509 -key cakey.pem -out cacert.pem -days 365

Se indica que es válido por un año, se introduce la contraseña cuando la solicite y se llena la información que se pide. Se generará el certificado cacert.pem en la carpeta donde está instalado OpenSSL.

Luego se crea el certificado de nuestro servidor web, a partir del fichero de solicitud del certificado (certreq.txt), de la clave privada (cakey.pem) y el certificado (cacert.pem) de la entidad certificadora. Para ello se debe colocar el fichero de la solicitud del certificado en la carpeta donde se encuentra OpenSSL y se escribe en la ventana de comandos de Windows la siguiente línea:

openssl x509 -req -days 365 -in certreq.txt -CA cacert.pem -CAkey cakey.pem -CAcreateserial -out iis.crt

Se genera en la carpeta mencionada un fichero denominado iss.crt.

Para añadir este certificado, se procede como se indicó anteriormente cuando se terminó la explicación de cómo se genera una solicitud de certificado. Se llegaría a un cuadro que muestra el resumen de la información, y luego se finalizará la instalación del certificado.

Para comprobar que todo esté funcionando correctamente, y ver que se tiene un servidor Web seguro, se puede acceder al navegador de la siguiente manera https://localhost. Se mostrará un cuadro, que indica que se está en presencia de un servidor seguro, pero que no reconoce la entidad certificadora.

CONCLUSIONES

- Las nuevas versiones perfeccionadas y estables de SSL permiten asegurar que es el protocolo por excelencia para garantizar la seguridad en el intercambio de información entre cliente y servidor en la web. Constituye unos de los mecanismos de seguridad más utilizados en la actualidad, viene incorporado a la mayoría de los navegadores.
- SSL es independiente de la aplicación que lo utilice, es decir, que no solo puede ser utilizado para ofrecer seguridad en las comunicaciones HTTP, sino también en aplicaciones como IMAP, FTP, Telnet.
- La instalación de SSL depende del servidor web a emplear pero en cualquier caso, poseer un certificado digital es de vital importancia para la garantía en la seguridad del protocolo.

REFERENCIAS

1. *El mercado europeo de certificados SSL está creciendo*, http://tienda.cyberplanet.es/ssl.aspx (consultado: julio, 2005).
2. *Comercio electrónico: Definición y evolución*, http://html.rincondelvago.com/comercio-electronico_definicion-y-evolucion.html (consultado: julio, 2005).
3. *uServer, SSL,* http://web.uservers.net/soporte/doc-print.php?articulo = 87 (consultado: septiembre, 2005).
4. **SÁNCHEZ GUERRERO, M. LOURDES; MARIAN HENAINE ABED Y SILVIA GONZÁLEZ BRAMBILA:** "*Firewall y algoritmos de seguridad en el Web*, http://ccc.inaoep.mx/~edominguez/MARS/Sanchez_SSL.doc (consultado: julio, 2005).
5. **RUZ, MIGUEL A.:** *Delitos Informáticos. Protocolo SSL*, http://www.delitosinformaticos.com (consultado: octubre, 2005).
6. *Netscape Communication Corporation. SSL version 3.0,* http://wp.netscape.com/eng/ssl3/ (consultado: diciembre, 2005).
7. *SSL: Introduction to Secure Sockets Layer*, http://www.cisco.com/en/US/netsol/ns340/ns394/ns50/ns140/networking_solutions_white_paper09186a0080136858.shtml (consultado: julio, 2005).
8. *Investigations about SSL*, Octubre, 2001 http://www.eucybervote.org/Reports/MSI-WP2-D7V1-V1.0-02.htm (consultado: julio, 2005).
9. *SSL v3.0 Specification*, http://wp.netscape.com/eng/ssl3/3-SPEC.HTM#7-6-8 (consultado: julio, 2005).

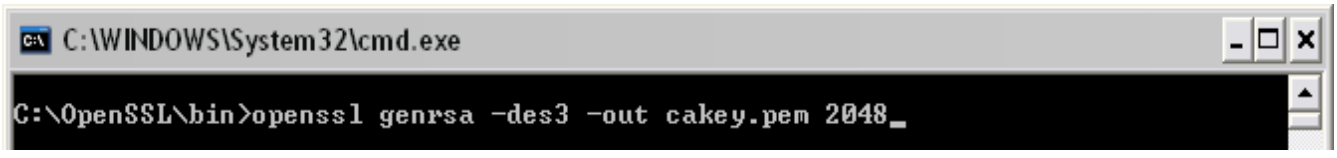

Fig. 3 *Pantalla del primer paso para generar una clave privada de la CA.*