Consejería de Hacienda y Administración Pública

Guía de incidencias del cliente

Sevilla, noviembre de 2010

Hoja de control		
Fecha	**Autor**	**Descripción**
03/11/2010	RPV	Adaptación guía de incidencias del cliente MPR

1 Introducción 4
2 Objetivos 4
3 Introducción 5
4 Incidencias conocidas del núcleo del cliente @firma 5
4.1 Incidencias generales 5
4.2 Instalación del Cliente 12
4.3 Despliegue del Cliente 14
4.4 Firmas Generales 15
4.5 Firma Web 16
4.6 Sobres Digitales 16
4.7 Incidencias específicas de la plataforma Windows 16
4.8 Incidencias específicas de la plataforma Linux / Sun Solaris 18
4.9 Incidencias específicas de la plataforma Mac OS X 18
4.10 Incidencias específicas de las firmas PDF 19
4.11 Incidencias específicas de las firmas XML 19
5 Glosario de términos 21

1 Introducción

El Cliente de Firma es una herramienta de Firma Electrónica que funciona en forma de Applet de Java integrado en una página Web mediante JavaScript.

El Cliente hace uso de los certificados digitales X.509 y de las claves privadas asociadas a los mismos que estén instalados en el repositorio o almacén de claves y certificados (*keystore*) del navegador web (*Internet Explorer, Mozilla, Firefox*) o el sistema operativo así como de los que estén en dispositivos (tarjetas inteligentes, dispositivos *USB*) configurados en el mismo (el caso de los DNI-e).

El Cliente de Firma, como su nombre indica, es una aplicación que se ejecuta en cliente (en el ordenador del usuario, no en el servidor Web). Esto es así para evitar que la clave privada asociada a un certificado tenga que "salir" del contenedor del usuario (tarjeta, dispositivo USB o navegador) ubicado en su PC. De hecho, nunca llega a salir del navegador, el Cliente le envía los datos a firmar y éste los devuelve firmados.

El Cliente de Firma contiene las interfaces y componentes web necesarios para la realización de los siguientes procesos (además de otros auxiliares como cálculos de hash, lectura de ficheros, etc…):

- Firma de formularios Web.
- Firma de datos y ficheros.
- Multifirma masiva de datos y ficheros.
- Cofirma (CoSignature) → Multifirma al mismo nivel.
- Contrafirma (CounterSignature) → Multifirma en cascada.

Como complemento al cliente de firma, se encuentra un cliente de cifrado que nos permite realizar las funciones de encriptación y desencriptación de datos atendiendo a diferentes algoritmos y configuraciones. Además permite la generación de sobres digitales.

2 Objetivos

El objetivo del presente documento es enumerar las dificultades típicas que pueden encontrar los integradores o sus usuarios durante la instalación, despliegue, integración o uso del Cliente @firma, así como las vías de resolución o paliación de estas.

3 Introducción

Este documento detalla la solución a ciertos problemas de instalación o ejecución del Cliente de Firma del MPR que, en algunas ocasiones, requiere de algunas actuaciones directamente sobre la máquina virtual, o bien, sobre el directorio donde se instala el cliente.

Para ello, a continuación detallamos los errores conocidos más importantes que se han localizado hasta el momento en el Cliente de Firma.

4 Incidencias conocidas del núcleo del cliente @firma

4.1 Incidencias generales

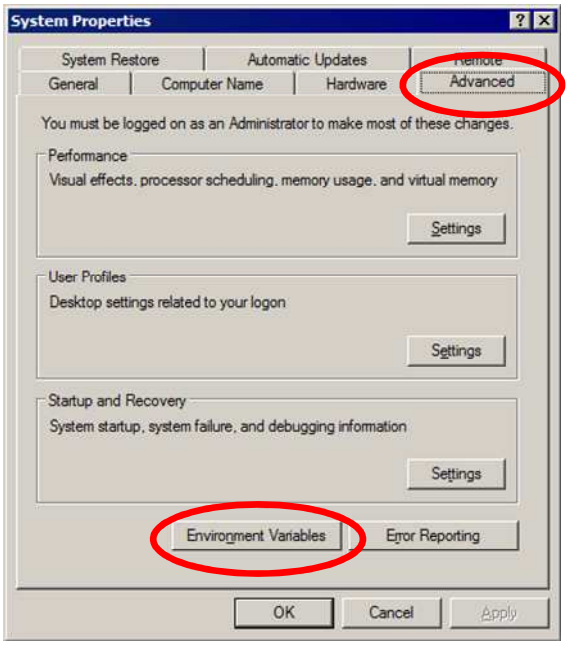
El cliente informa de que no se han encontrado certificados en el almacén de Mozilla Firefox, pese a que hay al menos un certificado instalado con clave privada y apto para firmas electrónicas.

El acceso al almacén de certificados de Mozilla Firefox es muy sensible a la versión de las bibliotecas NSS (*Netscape Security Services*), por lo que en caso de tener diferentes revisiones instaladas, puede estar cargándose la versión equivocada.

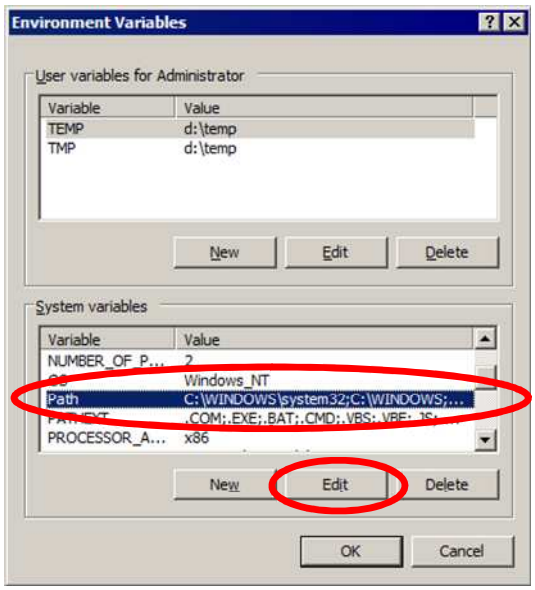
Para evitar este inconveniente es necesario asegurarnos de que en las variables de entorno que indican al sistema operativo la precedencia de directorios en la carga de bibliotecas (PATH en Windows, PATH y `LD_LIBRARY_PATH` en sistemas operativos basados en UNIX), el directorio que contiene la versión de las bibliotecas NSS correspondiente a la versión de Mozilla Firefox que ejecuta el Cliente @firma, debe ser el primero en prioridad.

En Microsoft Windows, el modo de realizar este cambio es el siguiente:

1. Seleccionar el icono "Sistema" (Windows 2000 y XP, consulte con la documentación de sus sistema operativo para otras versiones) en el Panel de Control, y abrir la pestaña "Avanzado" en el diálogo que aparece. Dentro del panel correspondiente a esa pestaña, pulsar en el botón "Variables de Entorno".

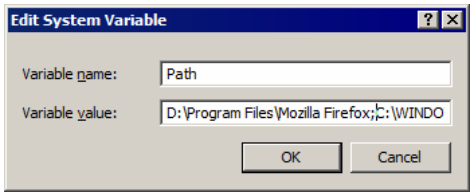


2. En el nuevo diálogo que aparece, debemos editar la variable Path, seleccionándola en la lista y pulsando el botón "Editar". Esta variable normalmente se encuentra en el grupo de variables del sistema, pero si se detectase que también están en variables del usuario (grupo de la parte superior del diálogo) deben editarse ambas de la misma manera.



3. Ya en el diálogo de edición, debemos insertar (nunca eliminar el valor actual) al principio del cuadro de texto de valor de la variable el directorio donde se encuentren los binarios del navegador Mozilla Firefox (donde encontremos el fichero `firefox.exe`, normalmente

"C:\Archivos de Programa\Mozilla Firefox") seguido de un punto y coma (";"), que actúa de separador entre los otros directorios ya existentes.



4. Tras aceptar los cambios (pulsando los botones "Aceptar" hasta cerrar todos los diálogos), reinicie su computadora.

En caso de que el problema persista, es posible que se esté cargando alguna de las dependencias de NSS desde alguno de los directorios del sistema, como por ejemplo desde el directorio "system32" en sistemas Windows. De ser el caso, deberán eliminarse manualmente las siguientes dependencias de NSS de estos directorios:

- libnspr4.dll
- libplc4.dll
- libplds4.dll
- nss3.dll
- nssckbi.dll
- smime3.dll
- softokn3.dll
- ssl3.dll

En caso de haber instalado anteriormente el Cliente @firma 2.4 o anteriores en sistemas Windows, es posible encontrar estas bibliotecas en el directorio "C:/Windows/system32" (la letra de la unidad puede variar según su sistema).

Cuando se recuperan desde Java ficheros XML en formato Base64 como resultado de operaciones de firma la codificación de caracteres se corrompe.

Durante la creación de un *String* de Java a partir de un binario obtenido a su vez de la decodificación de un Base64 se pueden pervertir los caracteres especiales de los ficheros XML si se indica una codificación errónea en el constructor de la clase *String*. La solución más rápida es no indicar codificación y confiar en las capacidades de Java de auto-detección de formato de caracteres. Si esta auto-detección de Java sigue proporcionando resultados incorrectos siempre puede obtener los XML directamente como texto

en vez de en Base64 usando el método `getSignatureText()` en vez de `getSignatureBase64Encoded()`.

El Cliente no completa correctamente las operaciones de firma cuando se ejecuta sobre Java 5, indicando en la consola de Java que se lanzaron ciertas excepciones.

El cliente necesita, dentro de la rama Java 5, al menos la versión 1.5u18 (se recomienda encarecidamente la actualización a Java 6u19 o al menos Java 5u22). Si está usando versiones de Java anteriores a 1.5u18 actualice su entorno de ejecución de Java (JRE) a una versión más moderna.

El Cliente, cuando se ejecuta sobre Java 5 actualiza algunas bibliotecas del propio entorno de ejecución ¿Por qué? ¿Puede tener alguna repercusión sobre otras aplicaciones Java?

El cliente actualiza los API Apache Xalan y Apache Xerces de Java 5 por las últimas versiones disponibles a fecha de publicación de este. Estas versiones son completamente compatibles con las anteriores incluidas con Java 5, por lo que no introducen ningún problema de compatibilidad.

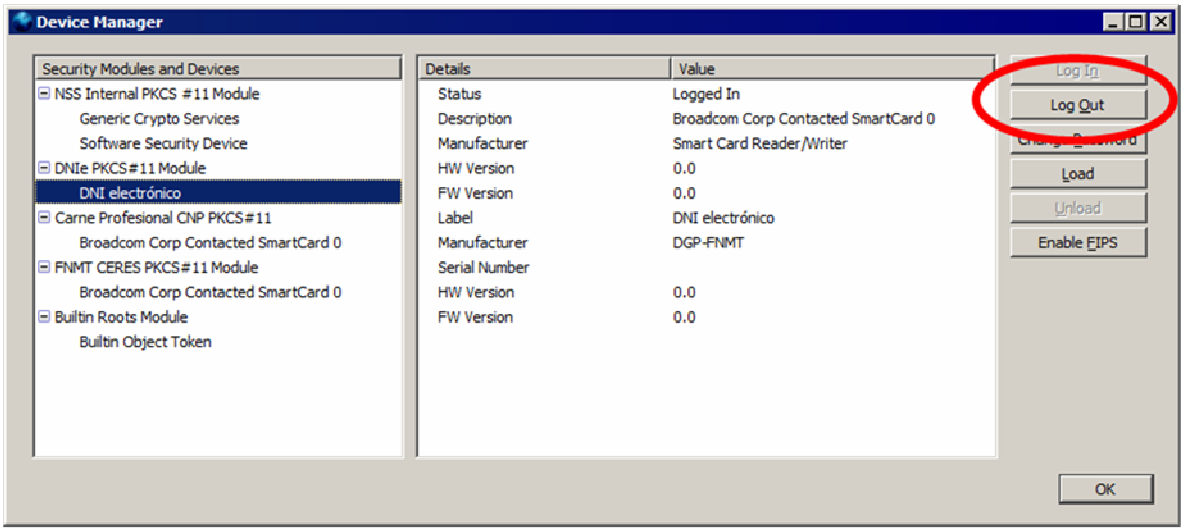
Adicionalmente, si se detecta la versión 5 de JRE se instala el proveedor de seguridad SunMSCAPI en su versión 6, ya que Java 5 originalmente no lo incluye. Esta instalación no cambia ni actualiza ninguna funcionalidad, sino que añade posibilidades completamente nuevas, por lo que no es posible que suponga problema de compatibilidad alguno.

En ciertas ocasiones, usando el Cliente en Mozilla / Firefox con DNIe (DNI Electrónico) el cliente se queda bloqueado y no muestra el diálogo de selección de certificados, desbloqueándose si retiro el DNIe del lector

El controlador PKCS#11 del DNIe no admite que se establezcan varias sesiones de forma simultánea, y si por cualquier razón (sesión SSL, etc.) el propio navegador Web Mozilla / Firefox tiene ya establecida una comunicación con el DNIe en el momento en el que el Cliente @firma también lo necesita, este último se queda bloqueado esperando a que en navegador Mozilla / Firefox cierre su sesión. El cierre de la sesión contra el DNIe por parte de Mozilla / Firefox puede tardar varios **minutos** si el usuario no interviene, por lo que conviene forzar manualmente este cierre:

- Extraer el DNIe del lector y volverlo a insertar justo en el momento en el que se solicita la contraseña del Repositorio Central de certificados de Mozilla Firefox (antes de introducirla). Es posible que Mozilla / Firefox reabra la sesión en la reinserción (adelantándose al Cliente @firma), por lo que quizás necesite repetir la operación.
- Podemos indicar a Mozilla / Firefox que cierre la sesión pulsando el botón "Log out" teniendo el dispositivo "DNIe PKCS#11 Module" seleccionado en la ventana "Dispositivos de Seguridad" del menú Opciones de Mozilla Firefox. Al igual que en el método anterior, a

veces es necesario repetir la operación varias veces, ya que Mozilla / Firefox reabre automáticamente la comunicación con el DNIe sin dar tiempo al Cliente @firma a utilizarlo. En otras ocasiones, el botón aparece deshabilitado aunque Mozilla / Firefox tenga una sesión abierta contra el dispositivo, con lo que no es posible aplicar este método.



Este problema se da predominantemente en Linux, Solaris y Mac OS X. No se ha detectado en ningún caso en ninguna versión de Windows.

Una solución alternativa en sistemas basados en UNIX (Linux, Solaris, Mac OS X) es modificar la configuración de OpenSC (producto en el que se basa el controlador PKCS#11 del DNIe en estas plataformas indicando que nunca se debe bloquear el acceso a las tarjetas inteligentes.
Para realizar esta indicación debe modificar el archivo de configuración de OpenSC, normalmente situado en `/etc/opensc/opensc.conf` y asegurarse de que contiene una línea descomentada con la opción **`lock_login = false;`** :

```
# By default, the OpenSC PKCS#11 module will lock your card
# once you authenticate to the card via C_Login.
# This is to prevent other users or other applications
# from connecting to the card and perform crypto operations
# (which may be possible because you have already authenticated
# with the card). Thus this setting is very secure.
#
# This behavior is a known violation of PKCS#11 specification,
# and is forced due to limitation of the OpenSC framework.
#
# However now once one application has started using your
# card with C_Login, no other application can use it, until
# the first is done and calls C_Logout or C_Finalize.
```

```
# In the case of many PKCS#11 application this does not happen
# until you exit the application.
#
# Thus it is impossible to use several smart card aware
# applications at the same time, e.g. you cannot run both
# Firefox and Thunderbird at the same time, if both are
# configured to use your smart card.
#
# Default: true
lock_login = false;
```

Dado que este cambio puede tener implicaciones de seguridad con otras tarjetas inteligentes (la seguridad del DNIe no se ve comprometida por él, dado que implementa medidas adicionales de protección, como la implementación de la normativa CWA-14890), realice únicamente estas modificaciones si está completamente seguro de sus implicaciones.

En ciertas distribuciones de Linux (como Guadalinex v6) el cambio no tienen ningún efecto sobre los bloqueos con DNIe, por lo que no solucionará el problema).

En algunas ocasiones, se ha detectado que este cambio en la configuración de OpenSC afecta a la comunicación con la tarjeta inteligente, provocando que el PIN de la tarjeta llegue corrupto. Asegúrese de introducir correctamente el PIN de su tarjeta y, en caso de que vuelva a solicitarse, cancele la operación y deshaga el cambio en la configuración para evitar que repetidos intentos bloqueen su DNIe.

La Web donde está instalado el Cliente solicita certificado cliente, y aunque este funciona correctamente en Internet Explorer y otros navegadores, no ocurre lo mismo con Mozilla / Firefox

Consulte las sección 17.1 del manual del integrador para más información de cómo resolver este problema de configuración de Mozilla / Firefox.

El Cliente deja de funcionar tras ejecutar la Aplicación Web de firma de la Ventanilla Única de la Seguridad Social

Este aplicativo de Ventanilla Única de la Seguridad Social modifica partes del JRE reemplazando bibliotecas vitales para el Cliente @firma por versiones ya obsoletas.

En caso de que necesite inter-operar el Cliente @firma con la aplicación de Ventanilla Única de la Seguridad Social, por favor, abra una incidencia contra esta última.

Problema en la codificación a Base64 de ficheros grandes

En determinados entornos de usuario, existe un problema en la codificación de ficheros a Base 64 que lleva al cierre abrupto de la máquina virtual de Java y el subsiguiente error en la ejecución del propio cliente. Este problema surge al utilizar los siguientes métodos del cliente para la conversión de ficheros grandes a base 64:

- getFileBase64Encoded(boolean showProgress);
- getFileBase64Encoded(boolean showProgress, String strUri);

Aunque el tamaño máximo permitido para los ficheros ronda alrededor de los 4 Megabytes, este depende por entero del entorno de ejecución.

En el caso de necesitar este método para adjuntar ficheros a formularios web, se aconseja que se estime el tamaño de los ficheros que se adjuntarán y, en caso de superar este tamaño, se busquen métodos alternativos de carga. Por ejemplo, el guardado de los datos a disco (si no estuviesen ya) y la carga manual por parte del usuario a través de un componente "file" de HTML.

Pérdida de foco en ventanas

En ocasiones, las ventanas del cliente pierden el foco, haciendo imposible la interacción del usuario. Este error se debe a un error reconocido por Sun Microsystems a partir del JRE 1.5.0 que bloquea ciertas ventanas Java en Internet Explorer y Mozilla, perdiendo el foco y haciendo imposible la interacción con el usuario.

En muchos casos este error se solventa al cambiar el foco a otras ventanas, o minimizar/maximizar el navegador, para intentar que recupere el foco, aunque no siempre resulta efectivo, por lo que se deberá reiniciar el navegador y reintentar la operación. En caso de problemas graves con alguna aplicación Web concreta, es recomendable el uso de Internet Explorer, en donde el problema aparece en menor medida.

No se detecta la inserción/extracción del DNIe en el lector (u otra tarjeta inteligente)

A veces puede ocurrir que el navegador no detecta la extracción o introducción del DNIe (u otra tarjeta inteligente) en el lector, por lo que si no hemos introducido la tarjeta previamente a que se arranque el cliente de firma, no se encontrará el certificado. Otro posible caso es que una vez cargado el cliente, se extraiga la tarjeta y, al realizar una operación de firma, el navegador muestre los certificados de la tarjeta (aunque ya no esté presente) fallando al intentar utilizarlo.

Este es un problema del navegador en la gestión de los dispositivos criptográficos (PKCS#11 para Mozilla y CSP para Internet Explorer), que no informa a la sesión abierta en el almacén de certificados de los cambios que se producen en el mismo.

La solución más rápida al problema es el insertar la tarjeta antes de que se produzca la carga del cliente de firma.

No se detecta el certificado del DNIe tras una autenticación infructuosa

Cuando se introduce mal el PIN del DNIe, ocurre que el navegador no detecta sus certificados, incluso aunque posteriormente el usuario sí lo introduzca correctamente.

El problema viene del CSP (Cryptographic Service Provider) del DNI electrónico y la mejor forma de solucionarlo es extraer e insertar el DNIe en el lector de tarjeta y volverse a autenticar.

4.2 Instalación del Cliente

El Cliente de @firma, en su primera ejecución, copia ciertos ficheros al disco duro del usuario para garantizar una correcta ejecución. Estos ficheros son en gran mayoría bibliotecas binarias nativas, aunque si el usuario utiliza el entorno de ejecución la versión 5 de Java también se actualizan ciertas clases Java de este. Concretamente, la lista de ficheros instalados es la siguiente:

- Únicamente Java 5 (no se necesitan en versiones superiores)
  - Paquete de compatibilidad del cliente con Java 5 (Linux/Windows). Necesario para hacer uso desde Java 5 de las funcionalidades incorporadas en las versiones actuales de Java. Este paquete es obligatorio cuando se ejecuta el cliente desde esta versión de Java. Directorio de instalación: `$JAVA_HOME/lib/endorsed`
    - `afirma_5_java_5.jar`
  - Java MS-CAPI Native Library (solo se instala en sistemas Windows). Necesaria únicamente para el soporte de los almacenes de claves de Windows / Internet Explorer. Se encuentra en el fichero comprimido `capi.zip`. Directorio de instalación: `$JAVA_HOME/bin/`
    - `sunmscapi.dll`
  - Entorno de ejecución de Microsoft Visual C++ 7.1 (solo se instala en sistemas Windows). Necesaria únicamente para el soporte de los almacenes de claves de

Windows / Internet Explorer, es una dependencia derivada de la biblioteca anterior. Se encuentra en el fichero comprimido `msvcr71.zip`. Directorio de instalación: `$LIBRARY_PATH/`

- `msvcr71.dll`

- Java MS-CAPI Provider (solo se instala en sistemas Windows). Necesaria únicamente para el soporte de los almacenes de claves de Windows / Internet Explorer. Se encuentra en el fichero comprimido `mscapiJar.zip`. Directorio de instalación: `$JAVA_HOME/lib/ext/`

  - `sunmscapi.jar`

En los directorios de instalación, las siguientes cadenas representan directorios del sistema operativo dependientes de la instalación:

`$HOME`

Directorio de usuario (por ejemplo, `/export/home/user` en un sistema Linux o

`C:\Documents and Settings\user` en un sistema Windows)

`$JAVA_HOME`

Directorio de instalación del entorno de ejecución de Java

`$LIBRARY_PATH`

Directorio de bibliotecas del sistema (por ejemplo, `/lib` en un sistema Linux o `C:\Windows\SYSTEM32` en un sistema MS-Windows 32 Bits)

De los tres directorios, el primero no presenta necesidades especiales respecto a permisos, ya que el usuario siempre tiene los apropiados sobre él, pero los otros dos pueden estar restringidos a operaciones de lectura, ejecución o escritura, lo cual puede provocar una instalación fallida.

Dado que los directorios sujetos a necesidades de permisos son usados únicamente si el usuario utiliza la versión 5 del entorno de ejecución de Java, existen dos posibilidades para resolver los posibles fallos de instalación:

1. Actualización a Java 6 (solución recomendada).
2. Cambio de los permisos de usuario de los directorios afectados.
    a. Consulte el manual de usuario de su sistema operativo para los procedimientos de cambio de permisos en directorios.
3. Instalación manual de las bibliotecas.
    a. Debe descomprimir los ficheros comprimidos ZIP (consulte el manual de su sistema operativo para los procedimientos de descompresión de ficheros ZIP) en las carpetas apropiadas.
    b. Tras la descompresión debe igualmente ajustar los permisos de los directorios y las bibliotecas descomprimidas:
        i. Los directorios deben tener permisos de lectura, no es necesario permisos de escritura.
        ii. Las bibliotecas necesitan permisos de lectura y ejecución, no es necesario permisos de escritura.

4.3 Despliegue del Cliente

Cuando se despliega el Cliente en entornos donde las páginas HTML se generan dinámicamente no es posible cargar el Applet

Las páginas HTML provistas como ejemplo necesitan ciertos cambios cuando se quiere desplegar el Cliente en servidores donde las páginas se generan dinámicamente (como por ejemplo, Portlets en un Servidor de Portales):

- Las bibliotecas Java del cliente (JAR) deben situarse en una dirección estática dentro del servidor Web, como por ejemplo: *http://dirección/directorio_clases*
- El JavaScript (las bibliotecas JS) debe incluirse dentro de la página que invoca al Applet y puede generarse dinámicamente, pero debe editarse el fichero *constantes.js* para indicar su localización mediante una URL absoluta:

```
/*****************************************************************************
 * Ruta a los instalables.                                                   *
 * Si no se establece, supone que estan en el mismo directorio (que el HTML). *
 *****************************************************************************/
var baseDownloadURL = http://direccion/directorio_clases;

/*****************************************************************************
 * Ruta al instalador.                                                       *
 * Si no se establece, supone que estan en el mismo directorio (que el HTML).  *
 *****************************************************************************/
var base = http://direccion/directorio_clases;
```

4.4 Firmas Generales

Alguno de los formatos de firma generados con el Cliente @firma no validan adecuadamente en otras plataformas

Compruebe siempre las matrices de compatibilidad del cliente para verificar que los formatos no están sujetos a problemas de adecuación con normativas/estándares (cuando esto ocurra estará así indicado) y cuáles de los que no presentan estos problemas están soportados por su plataforma validadora.

Algunos dispositivos de creación de firma no funcionan con las funcionalidades de firma multi-fase del Cliente

Estas limitaciones son impuestas por los fabricantes de los dispositivos de creación de firmas y no es posible sortearlas. Consulte con el fabricante de su dispositivo de creación de firmas para comprobar que funcionalidades pueden estar restringidas.

La firma sin usar huella digital (algoritmo NONEwithRSA) no es capaz de firmar todos los datos que se le proporcionan

La firma sin huella digital NONEwithRSA necesita en determinados dispositivos que los datos de entrada sean de un tamaño y con un formato determinado. Este formato no se especifica en la documentación del cliente, ya que es dependiente del dispositivo de creación de firmas.

Evite el uso de NONEwithRSA si no está seguro de la compatibilidad de los datos de entrada con su dispositivo de creación de firmas.

Además, recuerde que ciertos usos de NONEwithRSA pueden resultar en firmas susceptibles de repudio según interpretaciones estrictas de las normativas.

La configuración de filtros de certificados produce un error cuando se establece un filtro de gran tamaño

Este error ocurre al usar un filtro de certificados mediante el método deprecado "`setCertFilter(String)`" o "`setMandatoryCertificateCondition(String)`". Al concatenar/anidar múltiples expresiones de este tipo se produce un error en la JVM que obliga a desactivar el filtro. Debe evitarse el uso de filtros con múltiples expresiones.

Es recomendable migrar las aplicaciones al nuevo sistema de filtros basado en la RFC 2254. Pueden establecerse filtros de este tipo mediante el método "`setCertFilterRFC2254(String, String, boolean)`".

4.5 Firma Web

No es posible firmar una página Web con el formato XMLDSig/XAdES Enveloped

XAdES/XMLDSig Enveloped solo admite firmar ficheros XML, y no todas las páginas HTML son compatibles XML.

Compruebe si las páginas HTML que desea firmar cumplen estrictamente con el formato XHTML (que sí es compatible XML) y si no seleccione otro formato de firmas.

4.6 Sobres Digitales

El cliente no permite usar el DNIe ni otros dispositivos seguros de creación de firmas para abrir sobres digitales

Ciertos emisores de certificados residentes en dispositivos seguros de creación de firma limitan en origen el uso que se les puede dar a estos.

En el caso del DNIe (y otros dispositivos), no se permite su uso para abrir sobres digitales por no estar autorizado por el emisor para ese uso. Debe siempre evitar enviar sobres digitales a particulares si no está seguro de que sus certificados (en su parte privada) van a permitir posteriormente su apertura.

4.7 Incidencias específicas de la plataforma Windows

El Cliente no Funciona Correctamente con Windows 64 bits

Para el correcto funcionamiento del Cliente en entornos Windows 64 bits (XP, 2003 Server, Vista, 2008 Server, 7) es necesario instalar un entorno de ejecución de Java en 32 bits, y **no** una variante de 64 bits.

El soporte de Java 64 bits en Windows 64 bits está supeditado al soporte de Java /JRE de las versiones 64 bits de CAPI en el caso de Internet Explorer, Google Chrome, Apple Safari y Opera y a la existencia de una versión nativa de NSS (*Netscape Security Services)* compilada en 64 bits en el caso de Mozilla Firefox.

El Cliente no Funciona Correctamente en Windows sobre arquitectura IA64 (Intel Itanium)

La arquitectura IA64 en Windows no está soportada por el Cliente y no lo estará en un futuro próximo.

El Cliente no permite realizar firmas sin formato (PKCS#1 v1.5) con el algoritmo NONEwithRSA en Internet Explorer

El error 6578658 de Java, que afecta a todos los JRE hasta la fecha, afecta a esta posibilidad, por lo que no es posible hacer firmas sin formato (PKCS#1 v1.5) usando el algoritmo NONEwithRSA mediante certificados residentes en Internet Explorer / CAPI.

Esta limitación puede afectar a las firmas multi-fase.

Más información: http://bugs.sun.com/view_bug.do?bug_id=6578658

El Cliente deja de funcionar por completo cuando estoy utilizándolo a la vez que una aplicación nativa Windows que hace uso de una tarjeta inteligente

Algunas aplicaciones nativas de Windows que hacen uso de tarjetas inteligentes como el DNIe (aplicación de escritorio, controles ActiveX en páginas Web de Internet Explorer…) interfieren en el funcionamiento de las bibliotecas SunMSCAPI de Java para el uso de los certificados del sistema operativo. Esta interferencia provoca que cualquier intento de una aplicación Java de acceder al almacén de certificados de Windows cuando se tiene insertada la tarjeta inteligente en el lector mientras la otra aplicación esté también ejecutándose, genere un error interno en la máquina virtual de Java que cierra instantáneamente la aplicación afectada.

Este es un problema generado por aplicaciones nativas Windows que acceden a CAPI por medios no recomendados y por defectos de la biblioteca SunMSCAPI, encargada del acceso al almacén de certificados de Windows, que no puede ser tratado, que impiden operar cuando se realizan estos accesos no recomendados.

En general, debe intentar evitarse la situación en donde una aplicación utilice una tarjeta inteligente a la vez que se usa el cliente de firma. Para hacerlo, conviene separar el uso de las dos aplicaciones que

acceden a la tarjeta mediante la extracción y reinserción de la misma en el lector o simplemente cerrando el resto de las aplicaciones mientras se usa una de ellas.

Si llegase a producirse este error, es posible que necesite cerrar la aplicación Windows que produce la incompatibilidad y reiniciar la aplicación (página Web) que integra el cliente @firma.

4.8 Incidencias específicas de la plataforma Linux / Sun Solaris

El Cliente no detecta ningún certificado bajo Mozilla / Firefox

El Cliente @firma, cuando se ejecuta en Linux o Sun Solaris necesita que las bibliotecas NSS estén situadas en "*/usr/lib*", "*/lib*" o al menos dentro de uno de los directorios incluidos en la variable de entorno `LD_LIBRARY_PATH`.

El Cliente no funciona adecuadamente en Linux / Sun Solaris 64 bits

El cliente necesita que las bibliotecas NSS del sistema estén compiladas en la misma arquitectura nativa que el entorno de ejecución de Java, por lo que si ha instalado un JRE de 64 bits necesitará un NSS de 64 bits, y si el JRE es de 32 bits, NSS debe ser también de 32 bits.

El Cliente @firma no provee las bibliotecas NSS para Sun Solaris, sino que utiliza las presentes en el sistema operativo.

4.9 Incidencias específicas de la plataforma Mac OS X

El Cliente, bajo Mozilla / Firefox utiliza el almacén del sistema operativo, pero no el propio del navegador Web

Dado que para acceder al almacén de Mozilla / Firefox en Mac OS X y Java 64 bits (es el Java actualmente soportado en Mac OS X) es necesario una versión 64 bits nativa de NSS y esta no existe, se ha optado por usar el almacén de certificados del sistema operativo (Llavero de Mac OS X).

En el momento que la Comunidad Mozilla ponga a disposición una compilación nativa de 64 bits de NSS se actualizará esta funcionalidad.

El cliente no puede acceder al DNIe en Mac OS X

Mac OS X utiliza los controladores Tokend de las tarjetas inteligentes para acceder a ellas y, por desgracia, el DNIe carece de este tipo de controlador. Esto conlleva que Mac OS X no pueda acceder al DNI electrónico a través de su propio almacén de certificados, que es el utilizado por el cliente @firma.

El motivo del porqué el cliente no accede al DNIe a través del almacén de Mozilla Firefox, que sí tiene acceso a él, se explica en la incidencia "El Cliente, bajo Mozilla / Firefox utiliza el almacén del sistema operativo, pero no el propio del navegador Web".

Por otro lado, no es posible acceder al DNIe directamente a través de su PKCS#11 debido a que este, en su versión para Mac OS X está compilado en arquitectura universal y Java requiere obligatoriamente que las librerías nativas esté en su misma arquitectura.

Como solución alternativa, es posible utilizar el paquete SCA de OpenSC. Este paquete hace de puente, permitiendo al sistema manejar las bibliotecas PKCS#11 de las tarjetas inteligentes como si se tratasen de bibliotecas Tokend. Si se opta por aplicar esta solución debe tenerse en cuenta que:

- Es necesaria una versión de SCA compatible con el controlador PKCS#11 para Mac OS X del DNIe.
- Es necesaria la última versión de los controladores del DNIe disponibles.
- Debe instalarse siempre SCA antes que los controladores del DNIe.

4.10 Incidencias específicas de las firmas PDF

El Cliente no permite la firma de PDF con ciertos certificados

Las firmas de documentos PDF realizadas externamente (que es el método utilizado por el Cliente) tienen un tamaño máximo de octetos que pueden ocupar dentro del PDF.

Como la firma incluye la cadena de certificación completa, si esta es muy extensa puede llegar a agotarse este espacio y resultar en una firma inválida o corrupta. Si esto le ocurre, por favor, póngase en contacto con el servicio de atención a los usuarios del Cliente @firma enviando una copia de su certificado de firma y la cadena de confianza completa. **Tenga siempre mucho cuidado de no enviar jamás las partes privadas de los certificados**.

4.11 Incidencias específicas de las firmas XML

El Cliente no firma las hojas de estilo de los ficheros XML

Dado que las hojas de estilo de un XML pueden declararse de distintas formas, el cliente adopta distintas estrategias para cada forma de declaración y según la variante de firma.

Las formas de declarar una hoja de estilo y la forma de firmar el XML en ese caso por el Cliente son las siguientes:

- La hoja de estilo está empotrada dentro del XML, y se declara con una referencia local (el valor de el atributo `href` de la declaración del XSL es un nombre de identificador de nodo XML precedido por "#").
  - En este caso no es necesaria ninguna estrategia adicional, pues al ser parte la hoja de estilo del XML, siempre que se firma uno, se firma también el otro. Esto aplica a la totalidad de las firmas XML.
- La hoja de estilo está accesible remotamente por protocolo HTTP o HTTPS (el valor del atributo `href` es una URL válida con esquema `http` ó `https`).
  - En este caso se añade una referencia a la firma que apunta a la hoja de estilo mediante la misma URL (una referencia Externally Detached). Esto aplica a la totalidad de las firmas XML).
- Se referencia a la hoja de estilo mediante una referencia relativa local.
  - En este caso, dado que las referencias relativas locales se pierden al firmar (el Applet no sabe en qué directorio o carpeta estaba el XML para localizar el XSL, y no puede asumir dónde se guardará la firma generada), las hojas de estilo no se firman.

Compruebe que el XML que desea firmar declara las hojas de estilo mediante alguno de los modos soportado.

Las firmas XMLDSig generadas no son compatibles con SOAP

Esta funcionalidad está en estudio para ser incluida en futuras versiones del Cliente.

Ciertos validadores no aceptan algunas de las firmas generadas por el Cliente @firma

El Cliente @firma permite generar firmas XML con configuraciones no reconocidas fuera de la Administración Pública Española (por ejemplo, las firmas explícitas XML), por lo que es posible que algunos productos de validación de firmas no las validen correctamente.

Revise con detalle la matriz de compatibilidad y las "NOTAS IMPORTANTES" del manual del Formato XML.

El Cliente no genera firmas XML usando huellas digitales SHA-2

El error de Java 6845600 (http://bugs.sun.com/view_bug.do?bug_id=6845600) afecta a la generación de firmas XML con SHA-256 y SHA-512, y el error de Java 6753664

(http://bugs.sun.com/view_bug.do?bug_id=6753664) impide usar estos algoritmos de huella digital sobre Internet Explorer.

En el momento de liberar la versión 1.6 u18 de Java, Sun Microsystems informó de que el error 6753664 había sido solucionado. Sin embargo, se comprobó que el problema persistía, por lo que se abrió el error 6946836 (http://bugs.sun.com/view_bug.do?bug_id=6946836) asociado a la incidencia.

5 Glosario de términos

Firma electrónica

Es el conjunto de datos, en forma electrónica, anejos a otros datos electrónicos o asociados funcionalmente con ellos, utilizados como medio para identificar formalmente al autor o a los autores del documento que la recoge.

XML Digital Signature (XMLDSig)

Es una recomendación del W3C que define una sintaxis XML para la firma digital

XML Advanced Signature (XAdES)

Es un conjunto de extensiones a las recomendaciones XML-DSig haciéndolas adecuadas para la firma electrónica avanzada.

RSA

Es un sistema criptográfico de clave pública desarrollado en 1977. En la actualidad, RSA es el primer y más utilizado algoritmo de este tipo y es válido tanto para cifrar como para firmar digitalmente.

XML

Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

Office Open XML (OOXML)

Es un formato de archivo abierto y estándar cuyas extensiones más comunes son .docx, .xlsx y .pptx. Se le utiliza para representar y almacenar hojas de cálculo, diagramas, presentaciones y documentos de

texto. Un archivo Office Open XML contiene principalmente datos basados en el lenguaje de marcado XML, comprimidos en un contenedor .zip específico.

Open Document Format (ODF)

Es un formato de fichero estándar para el almacenamiento de documentos ofimáticos tales como hojas de cálculo, memorandos, gráficas y presentaciones. Aunque las especificaciones fueron inicialmente elaboradas por Sun, el estándar fue desarrollado por el comité técnico para Open Office XML de la organización OASIS y está basado en un esquema XML inicialmente creado e implementado por la suite ofimática OpenOffice.org (ver OpenOffice.org XML).

ZIP

Es un formato de almacenamiento sin pérdida, muy utilizado para la compresión de datos como imágenes, programas o documentos.

PDF

Es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto).

SHA

Es un sistema de funciones hash criptográficas relacionadas de la Agencia de Seguridad Nacional de los Estados Unidos y publicadas por el National Institute of Standards and Technology (NIST). El primer miembro de la familia fue publicado en 1993 es oficialmente llamado SHA. Sin embargo, hoy día, no oficialmente se le llama SHA-0 para evitar confusiones con sus sucesores. Dos años más tarde el primer sucesor de SHA fue publicado con el nombre de SHA-1. Existen cuatro variantes más que se han publicado desde entonces cuyas diferencias se basan en un diseño algo modificado y rangos de salida incrementados: SHA-224, SHA-256, SHA-384, y SHA-512 (llamándose SHA-2 a todos ellos).

PKCS

Se refiere a un grupo de estándares de criptografía de clave pública concebidos y publicados por los laboratorios de RSA en California. A RSA Security se le asignaron los derechos de licenciamiento para la patente de algoritmo de clave asimétrica RSA y adquirió los derechos de licenciamiento para muchas otras patentes de claves.

W3C

Es un consorcio internacional que produce recomendaciones para la World Wide Web. Está dirigida por Tim Berners-Lee, el creador original de URL (Uniform Resource Locator, Localizador Uniforme de

Recursos), HTTP (HyperText Transfer Protocol, Protocolo de Transferencia de HiperTexto) y HTML (Lenguaje de Marcado de HiperTexto) que son las principales tecnologías sobre las que se basa la Web.

OpenOffice.org

Es una suite ofimática libre (código abierto y distribución gratuita) que incluye herramientas como procesador de textos, hoja de cálculo, presentaciones, herramientas para el dibujo vectorial y base de datos. Está disponible para varias plataformas, tales como Microsoft Windows, GNU/Linux, BSD, Solaris y Mac OS X. Soporta numerosos formatos de archivo, incluyendo como predeterminado el formato estándar ISO/IEC OpenDocument (ODF), entre otros formatos comunes. A febrero de 2010, OpenOffice soporta más de 110 idiomas.

Base64

Es un sistema de numeración posicional que usa 64 como base. Es la mayor potencia de dos que puede ser representada usando únicamente los caracteres imprimibles de ASCII. Esto ha propiciado su uso para codificación de correos electrónicos, PGP y otras aplicaciones. Todas las variantes famosas que se conocen con el nombre de Base64 usan el rango de caracteres A-Z, a-z y 0-9 en este orden para los primeros 62 dígitos, pero los símbolos escogidos para los últimos dos dígitos varían considerablemente de unas a otras. Otros métodos de codificación como UUEncode y las últimas versiones de binhex usan un conjunto diferente de 64 caracteres para representar 6 dígitos binarios, pero éstos nunca son llamados Base64.

ASN.1

Es una norma para representar datos independientemente de la máquina que se esté usando y sus formas de representación internas. Es un protocolo de nivel de presentación en el modelo OSI.

Autoridad de Certificación (CA)

Es una entidad de confianza, responsable de emitir y revocar los certificados digitales o certificados, utilizados en la firma electrónica, para lo cual se emplea la criptografía de clave pública. Jurídicamente es un caso particular de Prestador de Servicios de Certificación.

Certificado Digital

Es un documento digital mediante el cual un tercero confiable (una autoridad de certificación) garantiza la vinculación entre la identidad de un sujeto o entidad y su clave pública.

Infraestructura de Clave Pública (PKI)

Es una combinación de hardware y software, políticas y procedimientos de seguridad que permiten la ejecución con garantías de operaciones criptográficas como el cifrado, la firma digital o el no repudio de transacciones electrónicas.