Análisis funcional del Cliente @firma

Índice de contenidos

[1 Introducción 3](#_Toc318731012)

[1.1 Objetivo 3](#_Toc318731013)

[1.2 Alcance 3](#_Toc318731014)

[1.3 Descripción 4](#_Toc318731015)

[2 Requisitos funcionales 7](#_Toc318731016)

[3 Descomposición funcional 9](#_Toc318731017)

[3.1 Firma 9](#_Toc318731018)

[3.2 Cofirma 11](#_Toc318731019)

[3.3 Contrafirmar 13](#_Toc318731020)

[3.4 Firma masiva de hashes (Procedimiento deprecado) 15](#_Toc318731021)

[3.5 Multifirma masiva de ficheros 17](#_Toc318731022)

[3.6 Multifirma masiva programática 19](#_Toc318731023)

[3.7 Cifrado simétrico 21](#_Toc318731024)

[3.8 Descifrado simétrico 23](#_Toc318731025)

[3.9 Ensobrado digital 25](#_Toc318731026)

[3.10 Abrir sobre digital 26](#_Toc318731027)

[Anexo A. Algoritmos de cifrado 27](#_Toc318731028)

[Anexo B. Sobres digitales 29](#_Toc318731029)

# Introducción

## Objetivo

El objeto de este documento es realizar una descripción general y funcional del cliente de firma y cifrado de la Plataforma @Firma.

## Alcance

Descripción del cliente de firma y cifrado y de las funcionalidades atribuidas. Abarcará los siguientes aspectos:

* Descripción de los tipos de firma y formatos soportados por el cliente.
* Identificación y descripción de los procesos que permite realizar el cliente de firma.
* Identificación de las funcionalidades de cifrado, los algoritmos empleados y la usabilidad de estos.
* Identificación de los sobres digitales soportados y la usabilidad de estos.

## Descripción

El cliente de firma, llamado Cliente @firma, es un componente que se ejecuta en la máquina del usuario en forma de Applet y le permite realizar diversas operaciones criptográficas:

* Firma / multifirma electrónica de documentos.
* Procesos de firma / multifirma masiva.
* Cifrado / descifrado de datos.
* Generación y apertura de sobre digitales.

La firma electrónica es el equivalente electrónico a la firma manuscrita. El usuario puede seleccionar cualquier tipo de datos para firmar.

El origen de los datos puede ser tanto un fichero, datos en memoria y, para algunos formatos de firma electrónica, huellas digitales (hashes).

Los formatos soportados por el sistema de firma son:

* PKCS#7 / CMS
* CAdES-BES / CAdES-EPES
* XMLdSig
* XAdES-BES / XAdES-EPES
* PAdES –BES / PAdES-EPES (Firmas PDF)
* ODF
* OOXML

Las firmas en formato CMS, CAdES, XMLdSig y XAdES soportarán 2 modos de firma:

* Implícita: La firma generada va acompañada de los datos, normalmente incluidos dentro de la firma.
* Explicita: La firma generada no mantiene relación directa con los datos, salvo posiblemente una referencia externa a ellos o su huella digital.

Las firmas PAdES, ODF y OOXML son por naturaleza implícitas y no soportarán este tipo de configuración.

El algoritmo utilizado por defecto será SHA1withRSA, salvo cuando el formato de firma exija lo contrario. Para aquellos formatos que permitan configurar el algoritmo se soportarán:

* MD5withRSA.
* SHA1withRSA.
* SHA256withRSA.
* SHA384withRSA.
* SHA512withRSA.

Se permite seleccionar el certificado electrónico para la firma de cualquiera de los siguientes almacenes de certificados:

* Almacén de Windows / Internet Explorer.
* Almacén de Mozilla Firefox.
* Almacén de Apple.
* Ficheros PKCS#12 / PFX.
* Ficheros JKS.
* Dispositivos externos (configurados en alguno de los almacenes ya mencionados).

El almacén de claves por defecto viene dado por el navegador Web y el sistema operativo desde el que se ejecute el Cliente @firma. Se configurará siempre que sea posible el mismo almacén que utiliza el navegador en ese sistema.

La multifirma es el procedimiento que permite a varios usuarios firmar un único documento de datos y almacenar sus firmas en un mismo documento de firma. La multifirma podrá ser:

* Cofirma o firma paralela, en el cual todos los firmantes están al mismo nivel y no existe un orden prefijado
* Contrafirma o firma en cascada o jerárquica, en la que los firmantes están en distintos niveles y sí existe orden entre las firmas.

La multifirma masiva es el proceso que permite automatizar la firma o multifirma de grandes cantidades de datos, aplicando la configuración en cada una de las operaciones (mismo certificado de firma, mismo algoritmo…). La multifirma masiva dispone de dos medios de aplicación:

* El usuario puede especificar el directorio con los ficheros que se desean multifirmar y configurar la operación que se aplicará a las firmas.
* El usuario puede desarrollar un mecanismo programático que le permita multifirmar múltiples ficheros, datos o hashes de forma más sencilla que ejecutando la operación concreta sobre cada uno de ellos.

El Cliente @firma dispone también de un sistema de cifrado de datos. Todos los algoritmos contemplados por este sistema son simétricos, es decir, se utiliza la misma clave para cifrar y descifrar los datos. Esto supone que la clave siempre debe ser privada.

Los algoritmos soportados por el sistema de cifrado son:

* AES
* Alleged RC4
* 3DES
* DES
* Blowfish
* RC2
* PBEWithSHA1AndDESede
* PBEWithSHA1AndRC2\_40
* PBEWithMD5AndDES

Los algoritmos PBE (los 3 últimos listados) utilizan contraseñas que puede insertar el propio usuario, aunque no son válidos todos los caracteres para estas. El resto de algoritmos usan directamente claves de cifrado que tienen unos requisitos específicos para cada algoritmo por lo que el sistema deberá tener en cuenta dichos requisitos.

Los sobres digitales utilizan una combinación de algoritmos de cifrado simétrico y asimétrico que permiten cifrar datos de tal forma que sólo una serie de personas serán capaces de descifrarlos, lo que los hace idóneos para la transmisión de datos cifrados a otras personas.

Los distintos tipos de sobres digitales soportados se encuentran descritos en los anexos.

# Requisitos funcionales

Los requisitos funcionales que debe cumplir son los expuestos a continuación:

* El sistema debe ser multiplataforma, pudiendo ser ejecutado en Windows, Linux (Red Hat, Ubuntu), Solaris/ OpenSolaris y Mac OS X; así mismo se debe poder ejecutar, al menos, en Internet Explorer, Firefox, Safari y Chrome.
* El componente accederá por defecto a los certificados instalados en el navegador donde se ejecute.
* Se permitirá utilizar un almacén de claves distinto al por defecto. Si el almacén dispone de una clave de acceso, se le solicitará al usuario.
* Se pueden filtrar los certificados disponibles mediante filtros basados en la especificación RFC2254 o mediantes expresiones regulares, definiendo valores críticos para determinados campos.
* El componente de firma será un applet e implementará toda la funcionalidad de la firma en la máquina cliente, de forma que se ejecutará siempre el mismo componente independientemente del proceso de firma que se quiera lanzar (firma/multifirma de ficheros, masiva…).
* El componente no instalará nada en la máquina del usuario salvo que sea absolutamente necesario y siempre pedirá confirmación al usuario para llevar a cabo este proceso. El cliente podrá ser cacheado por la máquina virtual y esta caché se actualizará automáticamente cada vez que se detecte una actualización desplegada en servidor.
* El componente firmará los datos introducidos, utilizando el certificado que seleccione el usuario, mediante el algoritmo “RSA” configurado y con un formato preestablecido.
* Se permitirá la generación de firmas EPES al configurar un formato de firma avanzada (CAdES, XAdES y PAdES) y los datos de una política de firma.
* El componente permitirá realizar firmas en serie (cofirmas) y en cascada (contrafirma) como una operación posterior a la operación de firma.
* Las contrafirmas podrán realizarse sobre firmas concretas del documento de firmas, sobre todas las firmas de uno o varios firmantes, sobre todas las firmas que no hayan sido anteriormente contrafirmadas o sobre todas las firmas.
* Se pueden guardar las firmas/multifirmas generadas en ficheros “.csig” (CMS y CAdES), “.xsig” (XADES y XMLDSIGN) y “.sig” (firma electrónica simple, sin formato) o incrustadas en los documentos firmados (PDF/PAdES, ODF y OOXML).
* Las multifirmas masivas se pueden realizar sobre un grupo de ficheros o sobre un conjunto de hashes. También podrán realizarse definiendo un proceso programático simple.
* El sistema de cifrado permitirá cifrar y descifrar contenidos en el cliente mediante algoritmos simétricos.
* Los algoritmos de cifrado disponibles serán: AES, Allleged RC4, Blowfish, TripleDES, DES, RC2, PBEWithSHA1AndDESede, PBEWithSHA1AndRC2\_40 y PBEWithMD5AndDES.
* Se permitirá generar automáticamente claves de cifrado para cada algoritmo de cifrado que lo soporte.
* Se podrán almacenar las claves de cifrado utilizadas en un almacén de claves único del usuario y utilizar estas claves para el cifrado y descifrado de datos.
* Se da la posibilidad de crear un sobre digital CMS para transmitir mensajes cifrados a uno o varios receptores.
* La incorporación de destinatarios a un sobre digital se deberá hacer mediante ficheros “.cer”, cargando el certificado en memoria o desde alguno de los almacenes del sistema.
* Será posible extraer certificados con clave pública de servidores LDAP simples.
* Se permite guardar en ficheros los datos generados en cada momento: firmas, ficheros, cifrados y descifrados, sobres digitales…

# Descomposición funcional

Una vez establecidos los requisitos funcionales, se van a identificar y definir cada uno de los procesos que deberá implementar para cumplir con dichos requisitos.

## Firma

Es el proceso mediante el cual el componente cliente firma unos datos utilizando la clave privada asociada a uno de los certificados que tiene el usuario instalado en su navegador.

El Cliente mostrará al usuario una lista de los certificados instalados en el almacén o dispositivo configurado y este deberá seleccionar uno de ellos. También podrá configurarse que se filtren previamente estos certificados e, incluso, que se seleccione directamente uno de ellos sin que el usuario tenga que elegirlo.

A continuación se detalla el caso de uso asociado a esta operativa:



##### Actores

* **Usuario:** Usuario que quiere realizar la firma.

##### Casos de Uso

* **Configurar firma:** Es el proceso en el que se establecen los distintos parámetros que afectará a la operación de firma.
  + Conf. Almacén: Configuración del almacén del que se extraerá el certificado de firma.
  + Conf. Filtro: Configuración de un filtro que restringirá los certificados mediante los que se podrá firmar. Se podrá configurar que el certificado se seleccione directamente si es el único que cumple ese filtro o que se muestre un diálogo de selección para que el usuario tome la decisión de cuál usar. Los filtros se basarán en el uso de la RFC-2254 y los KeyUsage de los certificados y permitirán filtrar por:
    - Estructura del *X509Principal* del titular del certificado.
    - Estructura del *X509Principal* del emisor del certificado.
    - Utilidad declarada para el certificado (firma, autenticación, envoltura de claves,…).
  + Conf. Política: Configuración de la política de firma.
  + Conf. Algoritmo: Configuración del algoritmo de firma.
  + Conf. Formato: Configuración del formato de firma. Incluye el formato a utilizar y la configuración específica que admita este (tipos de datos firmados, contexto de firma, versión del formato…)
  + Conf. Modo: Configuración del modo de firma.
* **Seleccionar documento:** Es el proceso en el que se seleccionan los datos sobre los que se desea realizar la firma. Por defecto, se solicita seleccionar un fichero. Las alternativas son las siguientes:
* Configurar fichero: Establece un fichero sobre el que realizar la firma.
* Configurar datos: Establece los datos sobre los que realizar la firma.
* Configurar hash: Establece el hash sobre el que realizar la firma. La operación de firma no aplicará algoritmo de hash.
* **Seleccionar certificado:** Es el proceso de selección del certificado con el que se va a firmar. Las alternativas son:
* Configurar alias: El integrador establece el alias del certificado del almacén configurado que se debe utilizar.
* Seleccionar en diálogo de certificados: Muestra un diálogo para la selección del certificado por parte del usuario. En este diálogo sólo aparecerán los certificados que cumplan el filtro de certificados definido anteriormente.
* **Firmar:** Ejecuta la operación de firma conforme a los parámetros configurados anteriormente.
* **Obtener firma:** Se toma el resultado de la operación de firma. Las alternativas, no excluyentes, son:
* Extraer firma: Extrae la firma generada.
* Guardar firma: Almacena en disco la firma generada.

Resultado Error: En caso de producirse un error, el proceso **Firmar** establecerá el mensaje correspondiente y lo pondrá a disposición del integrador.

## Cofirma

Permite a varios usuarios firmar el mismo elemento.



##### Actores

* **Usuario:** Usuario que quiere realizar la cofirma.

##### Casos de Uso

* **Configurar firma:** Es el proceso en el que se establecen los distintos parámetros que afectará a la operación de firma.
* Conf. Almacén: Configuración del almacén del que se extraerá el certificado de firma.
  + Conf. Filtro: Configuración de un filtro que restringirá los certificados mediante los que se podrá firmar. Se podrá configurar que el certificado se seleccione directamente si es el único que cumple ese filtro o que se muestre un diálogo de selección para que el usuario tome la decisión de cuál usar. Los filtros se basarán en el uso de la RFC-2254 y los KeyUsage de los certificados y permitirán filtrar por:
    - Estructura del *X509Principal* del titular del certificado.
    - Estructura del *X509Principal* del emisor del certificado.
    - Utilidad declarada para el certificado (firma, autenticación, envoltura de claves,…).
* Conf. Política: Configuración de la política de firma.
* Conf. Algoritmo: Configuración del algoritmo de firma.
* Conf. Formato: Configuración del formato de firma. Incluye el formato a utilizar y la configuración específica que admita este (tipos de datos firmados, contexto de firma, versión del formato…)
* Conf. Modo: Configuración del modo de firma.
* **Seleccionar documento:** Es el proceso en el que se seleccionan los datos sobre los que se desea realizar la firma. Por defecto, se solicita seleccionar un fichero. En el proceso de cofirma, la selección del documento no es obligatorio cuando la firma que se selecciona en el paso siguiente contiene ya los datos que se firmaron originalmente.
* Configurar fichero: Establece un fichero sobre el que realizar la firma.
* Configurar datos: Establece los datos sobre los que realizar la firma.
* Configurar hash: Establece el hash sobre el que realizar la firma. La operación de firma no aplicará algoritmo de hash.
* **Seleccionar firma:** Es el proceso de selección del documento de firma al que se le desea agregar una nueva cofirma.
* Conf. Fichero firma: Configura el fichero de firma al que se le quiere agregar la cofirma.
* Conf. Datos firma: Configura la firma (en memoria) a la que se quiere agregar la cofirma.
* **Seleccionar certificado:** Es el proceso de selección del certificado con el que se va a firmar.
* Configurar alias: Establece el alias del certificado en el almacén configurado.
* Seleccionar en diálogo de certificados: Muestra un diálogo para la selección del certificado por parte del usuario.
* **Cofirmar:** Ejecuta la operación de cofirma conforme a los parámetros configurados anteriormente.
* **Obtener firma:** Se toma el resultado de la operación de firma.
* Extraer firma: Extrae la firma generada.
* Guardar firma: Almacena en disco la firma generada.

Resultado Error. En caso de producirse un error, el proceso **Cofirmar** establecerá el mensaje correspondiente y lo pondrá a disposición del integrador.

## Contrafirmar

Permite firmar a un usuario la firma de otro usuario. La contrafirma se puede realizar sobre: todas las firmas de un documento, las firmas todavía no contrafirmadas, las firmas de un firmante o determinadas firmas.



##### Actores

* **Usuario:** Usuario que quiere realizar la contrafirma.

##### Casos de Uso

* **Configurar firma:** Es el proceso en el que se establecen los distintos parámetros que afectará a la operación de firma.
* Conf. Almacén: Configuración del almacén del que se extraerá el certificado de firma.
  + Conf. Filtro: Configuración de un filtro que restringirá los certificados mediante los que se podrá firmar. Se podrá configurar que el certificado se seleccione directamente si es el único que cumple ese filtro o que se muestre un diálogo de selección para que el usuario tome la decisión de cuál usar. Los filtros se basarán en el uso de la RFC-2254 y los KeyUsage de los certificados y permitirán filtrar por:
    - Estructura del *X509Principal* del titular del certificado.
    - Estructura del *X509Principal* del emisor del certificado.
    - Utilidad declarada para el certificado (firma, autenticación, envoltura de claves,…).
* Conf. Política: Configuración de la política de firma.
* Conf. Algoritmo: Configuración del algoritmo de firma.
* Conf. Formato: Configuración del formato de firma. Incluye el formato a utilizar y la configuración específica que admita este (tipos de datos firmados, contexto de firma, versión del formato…)
* Conf. Modo: Configuración del modo de firma.
* **Seleccionar firma:** Es el proceso de selección del documento de firma al que se le desea agregar las contrafirmas.
* Conf. Fichero firma: Configura el fichero con los datos de firma al que se le quiere agregar las contrafirmas.
* Conf. Datos firma: Configura los datos de firma a los que se quiere agregar las contrafirmas.
* **Seleccionar certificado:** Es el proceso de selección del certificado con el que se va a firmar.
* Configurar alias: Establece el alias del certificado en el almacén configurado.
* Seleccionar en diálogo de certificados: Muestra un diálogo para la selección del certificado por parte del usuario.
* **Contrafirmar:** Ejecuta la operación de contrafirma conforme a los parámetros configurados anteriormente. La operación aplicará a una serie de nodos de firma.
* Cont. Árbol: Contrafirma todos los nodos.
* Cont. Hojas: Contrafirma los nodos hoja del árbol.
* Cont. Nodos: Contrafirma una serie de nodos.
* Cont. Firmantes: Contrafirma todos los nodos de una serie de firmantes.
* Conf. Nodos: Configura los nodos/firmantes que se desean contrafirmar.
* **Obtener firma:** Se toma el resultado de la operación de firma.
* Extraer firma: Extrae la firma generada.
* Guardar firma: Almacena en disco la firma generada.

Resultado Error. En caso de producirse un error, la tarea **Contrafirmar** establecerá el mensaje correspondiente y lo pondrá a disposición del integrador.

## Firma masiva de hashes (Procedimiento deprecado)

Permite firmar una serie de hashes.



##### Actores

* **Usuario:** Usuario que quiere realizar la firma masiva de hashes.

##### Casos de Uso

* **Configurar firma:** Es el proceso en el que se establecen los distintos parámetros que afectará a la operación de firma.
* Conf. Almacén: Configuración del almacén del que se extraerá el certificado de firma.
  + Conf. Filtro: Configuración de un filtro que restringirá los certificados mediante los que se podrá firmar. Se podrá configurar que el certificado se seleccione directamente si es el único que cumple ese filtro o que se muestre un diálogo de selección para que el usuario tome la decisión de cuál usar. Los filtros se basarán en el uso de la RFC-2254 y los KeyUsage de los certificados y permitirán filtrar por:
    - Estructura del *X509Principal* del titular del certificado.
    - Estructura del *X509Principal* del emisor del certificado.
    - Utilidad declarada para el certificado (firma, autenticación, envoltura de claves,…).
* Conf. Política: Configuración de la política de firma.
* Conf. Algoritmo: Configuración del algoritmo de firma.
* Conf. Formato: Configuración del formato de firma. Incluye el formato a utilizar y la configuración específica que admita este (tipos de datos firmados, contexto de firma, versión del formato…). Los únicos formatos que soportan esta operación son CMS, CAdES, XMLdSIG y XAdES.
* Conf. Modo: Configuración del modo de firma.
* **Seleccionar hashes:** Operación mediante la que se seleccionan todos los hashes que se desean firmar.
* **Seleccionar certificado:** Es el proceso de selección del certificado con el que se va a firmar.
* Configurar alias: Establece el alias del certificado en el almacén configurado.
* Seleccionar en diálogo de certificados: Muestra un diálogo para la selección del certificado por parte del usuario.
* **Firmar hashes:** Ejecuta la operación de firma sobre cada uno de los hashes configurados.
* **Obtener firmas:** Obtiene todas las firmas generadas a partir de los hashes introducidos.

Resultado Error. En caso de producirse un error, la tarea **Firmar hashes** establecerá el mensaje correspondiente y lo pondrá a disposición del integrador.

## Multifirma masiva de ficheros

Permite multifirmar los ficheros de un directorio. Esto es, permite configurar una operación de firma o multifirma, establecer una configuración de firma y aplicarlas sobre cada uno de los ficheros de un directorio o una selección de los mismos.



##### Actores

* **Usuario:** Usuario que quiere realizar la multifirma masiva de ficheros.

##### Casos de Uso

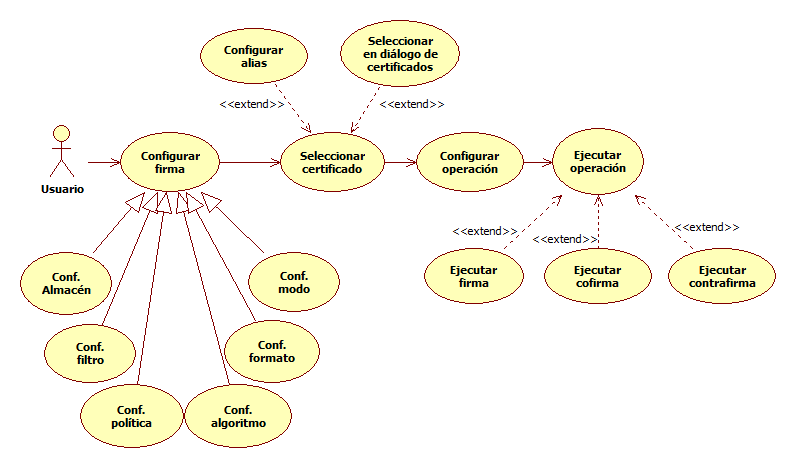
* **Configurar firma:** Es el proceso en el que se establecen los distintos parámetros que afectará a la operación de firma.
* Conf. Almacén: Configuración del almacén del que se extraerá el certificado de firma.
  + Conf. Filtro: Configuración de un filtro que restringirá los certificados mediante los que se podrá firmar. Se podrá configurar que el certificado se seleccione directamente si es el único que cumple ese filtro o que se muestre un diálogo de selección para que el usuario tome la decisión de cuál usar. Los filtros se basarán en el uso de la RFC-2254 y los KeyUsage de los certificados y permitirán filtrar por:
    - Estructura del *X509Principal* del titular del certificado.
    - Estructura del *X509Principal* del emisor del certificado.
    - Utilidad declarada para el certificado (firma, autenticación, envoltura de claves,…).
* Conf. Política: Configuración de la política de firma.
* Conf. Algoritmo: Configuración del algoritmo de firma.
* Conf. Formato: Configuración del formato de firma. Incluye el formato a utilizar y la configuración específica que admita este (tipos de datos firmados, contexto de firma, versión del formato…). Los únicos formatos que soportan esta operación son CMS, CAdES, XMLdSIG y XAdES.
* Conf. Modo: Configuración del modo de firma.
* Conf. Directorio salida: Configuración del directorio en el que se almacenarán las firmas generadas.
* Conf. Operación: Configuración de la operación a realizar: firma, cofirma, contrafirma de todo el árbol de firmas o contrafirma de los nodos hoja.
* Conf. Recursividad: Configuración de la recursividad de proceso. Si se indica que sea recursivo, se firmarán también los ficheros de los subdirectorios.
* **Seleccionar directorio:** Configuración del directorio con los ficheros que se desean firmar.
* **Seleccionar certificado:** Es el proceso de selección del certificado con el que se va a firmar.
* Configurar alias: Establece el alias del certificado en el almacén configurado.
* Seleccionar en diálogo de certificados: Muestra un diálogo para la selección del certificado por parte del usuario.
* **Firmar directorio:** Ejecuta la operación de masiva sobre cada uno de los ficheros del directorio. Las firmas resultantes irán a parar al directorio de salida o al mismo directorio de entrada si no se configuró. El resultado de las firmas individuales se reflejará en un fichero de log en el directorio de salida de las firmas.

Resultado Error. El proceso continúa hasta tratar todos los ficheros del directorio que sean objetivo de la operación. En caso de producirse errores en operaciones individuales del proceso, se reflejarán los mensajes correspondientes en el fichero de log generado.

## Multifirma masiva programática

Permite firmar, cofirmar y contrafirmar datos, fichero y hashes con una misma configuración de firma. Esta operación permitirá establecer una configuración de firma única y aplicarla en cada operación individual. El integrador aplicará la operación sobre cada objetivo de firma concreto, obteniendo directamente su resultado . Este proceso no se aplica a múltiples objetivos, el integrador debe ordenar explícitamente cada operación individual.

Este modo de operación está orientado a facilitar la integración del cliente en sistemas de firma masiva de documentos. Por ejemplo, recorrer un listado de datos objetivo y aplicar la operación sobre cada uno de ellos.



##### Actores

* **Usuario:** Usuario que quiere realizar la firma masiva de hashes.

##### Casos de Uso

* **Configurar firma:** Es el proceso en el que se establecen los distintos parámetros que afectará a la operación de firma.
* Conf. Almacén: Configuración del almacén del que se extraerá el certificado de firma.
  + Conf. Filtro: Configuración de un filtro que restringirá los certificados mediante los que se podrá firmar. Se podrá configurar que el certificado se seleccione directamente si es el único que cumple ese filtro o que se muestre un diálogo de selección para que el usuario tome la decisión de cuál usar. Los filtros se basarán en el uso de la RFC-2254 y los KeyUsage de los certificados y permitirán filtrar por:
    - Estructura del *X509Principal* del titular del certificado.
    - Estructura del *X509Principal* del emisor del certificado.
    - Utilidad declarada para el certificado (firma, autenticación, envoltura de claves,…).
* Conf. Política: Configuración de la política de firma.
* Conf. Algoritmo: Configuración del algoritmo de firma.
* Conf. Formato: Configuración del formato de firma. Incluye el formato a utilizar y la configuración específica que admita este (tipos de datos firmados, contexto de firma, versión del formato…). Los únicos formatos que soportan esta operación son CMS, CAdES, XMLdSIG y XAdES.
* Conf. Modo: Configuración del modo de firma.
* **Seleccionar certificado:** Es el proceso de selección del certificado con el que se va a firmar.
* Configurar alias: Establece el alias del certificado en el almacén configurado.
* Seleccionar en diálogo de certificados: Muestra un diálogo para la selección del certificado por parte del usuario.
* **Configurar operación:** Es el proceso por el que se configura el tipo de operación que se debe realizar. Las operaciones permitidas serán: firma, cofirma, contrafirma de nodos hoja y contrafirma del árbol de firma.
* **Ejecutar operación:** Ejecuta la operación configurada con los parámetros establecidos en los pasos anteriores. La operación devuelve directamente el resultado, no siendo necesario recogerlo en un paso posterior.

Resultado Error. En caso de producirse un error, la tarea **Ejecutar operación** no generará la firma y establecerá el mensaje correspondiente poniéndolo a disposición del integrador.

## Cifrado simétrico

Permite el cifrado de datos por medio de una clave simétrica. Esto es, se podrán descifrar los datos mediante esa misma clave.



##### Actores

* **Usuario:** Usuario que quiere realizar el cifrado.

##### Casos de Uso

* **Configurar cifrado:** Es el proceso en el que se establecen los distintos parámetros que afectarán a la operación de cifrado.
* Conf. Modo clave: Configuración del origen de la clave de cifrado (autogenerada, insertada por el usuario, mediante contraseña o desde almacén).
* Conf. Algoritmo: Configuración del algoritmo de cifrado.
* Conf. Clave/Contraseña: Configuración de la clave o contraseña de cifrado.
* Conf. Almacén cifrado: Configuración del almacén para extraer claves de cifrado.
* **Seleccionar documento:** Es el proceso en el que se selecciona el elemento que se desea cifrar. Por defecto, se solicita seleccionar un fichero.
* Configurar fichero: Establece un fichero a cifrar.
* Configurar datos: Establece los datos a cifrar.
* **Cifrar:** Ejecuta el cifrado conforme a los parámetros configurados anteriormente.
* **Obtener cifrado:** Se toma el resultado de la operación de cifrado.
* Extraer cifrado: Recupera los datos cifrados.
* Guardar cifrado: Almacena en disco los datos cifrados obtenidos.

Resultado Error. En caso de error, devuelve la cadena de texto asociada al error encontrado.

## Descifrado simétrico

Permite el descifrado simétrico de datos especificando la clave de cifrado.



##### Actores

* **Usuario:** Usuario que quiere descifrar los datos.

##### Casos de Uso

* **Configurar descifrado:** Es el proceso en el que se establecen los distintos parámetros que afectarán a la operación de descifrado.
* Conf. Modo clave: Configuración del origen de la clave para el descifrado (insertada por el usuario, mediante contraseña o desde almacén).
* Conf. Algoritmo: Configuración del algoritmo de cifrado.
* Conf. Clave/Contraseña: Configuración de la clave o contraseña de cifrado.
* Conf. Almacén cifrado: Configuración del almacén para la carga de claves de cifrado.
* **Seleccionar cifrado:** Es el proceso en el que se selecciona el documento cifrado que se desea descifrar. Por defecto, se solicita seleccionar un fichero.
* Configurar fichero: Establece la ruta del fichero cifrado.
* Configurar datos: Establece los datos cifrados.
* **Descifrar:** Ejecuta la operación de descifrado conforme a los parámetros configurados.
* **Obtener datos:** Se toma el resultado de la operación de cifrado.
* Extraer datos: Recupera los datos descifrados.
* Guardar datos: Almacena en disco los datos descifrados obtenidos.

Resultado Error. En caso de error, devuelve la cadena de texto asociada al error encontrado.

## Ensobrado digital

Permite generar un paquete de datos cifrados e indicar quienes podrán descifrarlo. Se utilizará una estructura CMS como soporte de los datos resultantes.



##### Actores

* **Usuario:** Usuario que quiere generar el sobre digital.

##### Casos de Uso

* **Configurar sobre:** Es el proceso en el que se establecen los distintos parámetros que afectarán a la generación del sobre digital.
* Conf. tipo sobre: Configuración del tipo de sobre que se desea generar: Sobre simple (EnvelopedData), sobre firmado (SignedAndEnvelopedData) o sobre autenticado (AuthenticatedAndEnvelopedData).
* **Seleccionar documento:** Es el proceso en el que se selecciona el elemento que se desea ensobrar. Por defecto, se solicita seleccionar un fichero.
* Configurar fichero: Establece un fichero a ensobrar.
* Configurar datos: Establece los datos a ensobrar.
* **Configurar destinatarios:** Configura quienes son los destinatarios del sobre digital a partir de sus certificados con clave pública. Estos certificados pueden tomarse de disco, de un almacén, de un servidor LDAP o configurarse directamente.
* **Configurar remitente:** Configura el remitente del sobre a partir de su certificado en almacén.
* **Ensobrar:** Genera el sobre digital conforme a los parámetros configurados anteriormente.
* **Obtener sobre:** Se toma el resultado de la operación de ensobrado.
* Recuperar sobre: Recupera el sobre digital.
* Guardar sobre: Almacena en disco el sobre digital.

Resultado Error. En caso de error, devuelve la cadena de texto asociada al error encontrado.

## Abrir sobre digital

Permite al receptor de un sobre digital extraer su contenido utilizando su propia clave privada.



##### Actores

* **Usuario:** Destinatario del sobre digital.

##### Casos de Uso

* **Seleccionar sobre:** Es el proceso en el que se selecciona el sobre digital del que se desean extraer los datos. Por defecto, se solicita seleccionar un fichero.
* Configurar fichero: Establece la ruta del sobre digital.
* Configurar datos: Establece el sobre digital.
* **Seleccionar destinatario:** Es el proceso de selección del certificado con el que se va a descifrar el sobre digital.
* Configurar alias: Establece el alias del certificado en el almacén configurado.
* Seleccionar en diálogo de certificados: Muestra un diálogo para la selección del certificado por parte del usuario.
* **Desensobrar:** Ejecuta la operación de extracción de datos del sobre digital.
* **Obtener datos:** Se toma el resultado de la operación de desensobrado.
* Extraer datos: Recupera los datos extraídos del sobre.
* Guardar datos: Almacena en disco los datos extraídos del sobre.

Resultado Error. En caso de error, devuelve la cadena de texto asociada al error encontrado.

1. Algoritmos de cifrado

Los algoritmos simétricos utilizan la misma clave tanto para cifrar como para descifrar, por lo que se recomienda una clave segura. Se incorporará un mecanismo de generación de claves seguras.

* **AES**: Advanced Encryption Standard. Sistema de encriptación adoptado por el gobierno de Estados Unidos (FIPS-PUB 197) y fue propuesto por el NIST. Cifrado simétrico en bloques de 128 bits y con claves de 128, 192 o 256 bits. Este estándar contempla el algoritmo para 10, 12 o 14 vueltas aunque está previsto aumentarlas debido al éxito obtenido con criptoanálisis que encuentra coincidencia con 277 bloques bajo 256 claves y requiere sobre 2224 pasos para completarse.
* **ARCFOUR**: Es el sistema de cifrado de flujo más utilizado y se usa en los protocolos más populares como Transport Layer Security (TLS/SSL) y Wired Equivalent Privacy (WEP) Este algoritmo fue excluido de los estándares de alta seguridad debido a su sistema de criptografía muy inseguro. El factor principal de su popularidad es su velocidad y simplicidad, así como la implementación tanto en software como en hardware es sencilla de desarrollar. El algoritmo es simple, ya que consiste en la combinación de 2 algoritmos 1-KSA y 2 PRGA
* **BLOWFISH**: Es un codificador de bloques simétricos, diseñado por Bruce Schneier en 1993. Usa bloques de 64 bits y claves que van desde los 32 bits hasta los 448 bits. Es un codificador de 16 rondas Feistel y usa llaves que dependen de las Cajas-S. Este algoritmo de cifrado correrá 521 veces para generar todas las subclaves, con lo que cerca de 4Kb de datos son procesados.
* **Data Encryption Standard (DES)**: Es un algoritmo prototipado del cifrado por bloques, un algoritmo que toma un texto de una longitud fija de bits y lo transforma mediante una serie de operaciones básicas en otro texto cifrado de la misma longitud. DES utiliza también una clave criptográfica para modificar la transformación, de modo que le descifrado sólo puede ser realizado por aquellos que conozcan la clave concreta utilizada en el cifrado. El cifrado se realiza con una clave de 56 bits, la cual es corta, y debido a algunos elementos de diseño clasificados propició las sospechas de la existencia de una puerta trasera para la National Security Agency.
* **Triple DES (3DES)**: Basado en DES (Data Encryption Standard). Cifrado en bloques de 64 bits con clave de 168 bits (192 incluyendo bits de paridad). Actualmente ha sido sustituido por AES, pero se sigue utilizando ampliamente. Los últimos criptoanálisis revelan que son necesarias 2108 operaciones para romper una clave TripleDES, pero si se concentra en las 3 claves DES, basta con 290 operaciones.
* **RC2**: Es un cifrado de bloques diseñado por Ron Rivest en 1987. El desarrollo de RC2 fue patrocinado por Lotus el cual fue exportado como parte de su software Lotus Notes. Este cifrado cuyo tamaño de clave es de 40 bits fue aprobado para exportarlo en 1989 bajo la regulación de los EEUU. RC2 es un cifrado de bloques de 64 bits con una clave de tamaño variable. Consta de 18 ciclos, 16 de un tipo MIXING y dos ciclos de otro tipo MASHING.
* **Algoritmos PBE (Password Basic Encryptation)**: Algoritmos que generan una clave secreta a partir de una contraseña introducida por el usuario. Estas técnicas plantean problemas debido a que los requerimientos de usuario difieren de los requerimientos de seguridad del sistema, ya que el usuario desea una frase fácil de recordar, sin embargo una clave segura actualmente requiere al menos 128 bits aleatorios. Además, los sistemas de encriptación basados en contraseñas son normalmente utilizados en aplicaciones donde un atacante puede repetidamente intentar adivinar la contraseña, con lo que se pueden utilizar métodos de fuerza bruta para obtener las contraseñas. Existen varios algoritmos que utilizan este sistema, como:
* **PBEWithSHA1AndRC2\_40**: Este modo encripta con 40-bits RC2
* **PBEWithMD5AndDES**: Este algoritmo combina los beneficios de una lenta e insegura encriptación de 56 bits con una insegura función hash. No es recomendable su uso. Es definido como “PKCS #5”
* **PBEWithSHA1AndDESede:** Este algoritmo utiliza una encriptación triple DES y una función de hash sha1.

1. Sobres digitales

El estándar CMS permite define una arquitectura capaz de encapsular distintos tipos de contenido como datos planos, datos firmados, datos cifrados, hashes… Las estructuras definidas por CMS nos permiten construir sobres digitales.

Nuestro objetivo al usar sobres digitales será poder distribuir documentos de forma segura y privada a uno o varios destinatarios. La estructura general del sobre será:



Donde ContentType será una cadena asociada ASN.1 que define el tipo del contenido y Content será los datos contenidos en el CMS. Estos datos los dividiremos en dos modos de implementación de CMS.

Debemos recordar que los tipos CMS son anidables y que, por ejemplo, el contenido a encriptar puede ser un contenido previamente firmado y viceversa.

CMS cifrado

Es un sobre digital simple que contiene el elemento cifrado y en el cual no se incluye ninguna información sobre la clave de cifrado. Esto hace que la clave de cifrado deba ser transmitida de forma ajena al envoltorio CMS. La estructura de este CMS es la siguiente:



Donde:

CMS versión será 0 si no existen atributos asociados y 2 si existen.

* *EncryptedContentInfo* contiene la información encriptada.
* ContentType: Define el tipo del contenido definida según ASN.1
* ContentEncryptionAlgorithm: Indica el algoritmo utilizado para encriptar el contenido.
* EncryptedContent: Contenido cifrado por el algoritmo anteriormente indicado y la clave suministrada a los receptores.
* *Attributes* es un conjunto de atributos no encriptados que definen algunos parámetros como el algoritmo utilizado.

CMS envuelto

Este tipo de sobre digital permite cifrar un contenido y distribuirlo a varios receptores e incluye la clave de cifrado. La estructura es la siguiente:



Donde encontramos los siguientes elementos:

* **CMS version**, indica la versión de CMS y estará definido según los parámetros OriginatorInfo, RecipientInfo y Attributes (definidos en RFC3852).
* **OriginatorInfo**. Incluye opcionalmente información sobre el creador del mensaje. Puede incluir una lista de certificados, con intención de crear un camino de certificación hasta el root, o una colección de CRLs.
* **RecipientInfo**. Incluye la información necesaria para que los destinatarios puedan descifrar el mensaje. Aunque existen varios tipos, el más útil para nuestro propósito es el que transporta la clave de cifrado y cuyos campos vienen definidos a continuación:
* CMS version. Define la versión del campo en función del tipo de identificador elegido.
* RecipientIdentifier. Puede incluir el emisor y número de serie del certificado (versión 0) o un identificador del certificado (versión 2).
* KeyEncryptionAlgorithm. Define el algoritmo utilizado para cifrar la clave.
* EncryptedKey. Clave usada para cifrar el contenido del documento y cifrada a su vez por el algoritmo especificado anteriormente.
* **EncryptedContentInfo.** Se compone de:
* ContentType: Define el tipo del contenido definido según ASN.1
* ContentEncryptionAlgorithm: Indica el algoritmo utilizado para cifrar el contenido.
* EncryptedContent: Contenido cifrado por el algoritmo anteriormente indicado y la clave suministrada a los receptores.

PKCS#7 firmado y envuelto

Este tipo de sobre no está definido dentro del estándar CMS sino de su predecesor, PKCS#7. Los PKCS#7 firmados y envueltos van firmados por quien genera el sobre y permiten cifrar un contenido y distribuirlo a varios receptores. La clave de cifrado del contenido va incrustada en el propio sobre. La estructura es la siguiente:

**SignedAndEnvelopedContentType**

**RecipientInfo {1, n}**

Version {1}

Version {1}

IssuerAndSerialNumber {1}

KeyEncryptionAlgorithm {1}

EncryptedKey {1}

**EncryptedContentInfo {1}**

Attributes {0, m}

ContentType {1}

ContentEncryptionAlgorithm {1}

EncryptedContent {1}

SignersInfo {0, l}

Donde encontramos los siguientes elementos:

* **Version**, indica la versión de la sintaxis con la que se ha construido el objeto.
* **RecipientInfo**. Incluye la información necesaria para que los destinatarios puedan descifrar el mensaje. Aunque existen varios tipos, el más útil para nuestro propósito es el que transporta la clave de cifrado y cuyos campos vienen definidos a continuación:
* Version. Indica la versión de la sintaxis con la que se ha construido el objeto.
* IssuerAndSerialNumber. Puede incluir el emisor y número de serie del certificado.
* KeyEncryptionAlgorithm. Define el algoritmo utilizado para cifrar la clave.
* EncryptedKey. Clave usada para cifrar el contenido del documento y cifrada a su vez por el algoritmo especificado anteriormente.
* EncryptedContentInfo. Se compone de:
* ContentType: Define el tipo del contenido definido según ASN.1
* ContentEncryptionAlgorithm: Indica el algoritmo utilizado para cifrar el contenido.
* EncryptedContent: Contenido cifrado por el algoritmo anteriormente indicado y la clave suministrada a los receptores.
* **SignerInfos.** Listado de firmas del contenido.

CMS Autenticado y envuelto

Este tipo de sobre viene definido en la RFC 5083, que actualiza el estándar CMS (RFC 3852). Este tipo de contenido combina los métodos de autenticación y encriptado de datos definidos en la RFC 3852. La estructura es la siguiente:

**AuthenticatedEnvelopedContentType**

**RecipientInfo {1, m}**

OriginatorInfo {1, n}

Version {1}

IssuerAndSerialNumber {1}

KeyEncryptionAlgorithm {1}

EncryptedKey {1}

**EncryptedContentInfo {1}**

Attributes {0,l}

ContentType {1}

ContentEncryptionAlgorithm {1}

EncryptedContent {1}

MessageAuthenticationCode, {1}

CMSVersion {1}

Donde encontramos los siguientes elementos:

* **Version**, indica la versión de la sintaxis con la que se ha construido el objeto.
* **OriginatorInfo**. Incluye información sobre el creador del mensaje. Puede incluir una lista de certificados, con intención de crear un camino de certificación hasta el root, o una colección de CRLs.
* **RecipientInfo**. Incluye la información necesaria para que los destinatarios puedan descifrar el mensaje. Aunque existen varios tipos, el más útil para nuestro propósito es el que transporta la clave de cifrado y cuyos campos vienen definidos a continuación:
* Version. Indica la versión de la sintaxis con la que se ha construido el objeto.
* IssuerAndSerialNumber. Puede incluir el emisor y número de serie del certificado.
* KeyEncryptionAlgorithm. Define el algoritmo utilizado para cifrar la clave.
* EncryptedKey. Clave usada para cifrar el contenido del documento y cifrada a su vez por el algoritmo especificado anteriormente.
* **EncryptedContentInfo.** Se compone de:
* ContentType: Define el tipo del contenido definido según ASN.1
* ContentEncryptionAlgorithm: Indica el algoritmo utilizado para cifrar el contenido.
* EncryptedContent: Contenido cifrado por el algoritmo anteriormente indicado y la clave suministrada a los receptores.
* **MenssageAuthenticationCode.** Código de autenticación del contenido.