Universitat Oberta de Catalunya uoc.edu

Servicio de firma de documentos privados

Nombre Estudiante Erick Ricardo Aramayo Monrroy
Máster Universitario en Seguridad de las Tecnologías de la Información y de la Información y las Comunicaciónes (MISTIC)
Sistemas de autenticación y autorización

Director del TFM: Juan Carlos Fernández Jara
Profesor/a responsable de la asignatura: Victor Garcia Font

29 de diciembre de 2020

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Servicio de firma de documentos privados*
Nombre del autor:	*Erick Ricardo Aramayo Monrroy*
Nombre del consultor/a:	*Juan Carlos Fernandez Jara*
Nombre del PRA:	*Victor Garcia Font*
Fecha de entrega (mm/aaaa):	12/2020
Titulación:	*MISTIC*
Área del Trabajo Final:	*Sistemas de autenticación y autorización*
Idioma del trabajo:	*Castellano*
Palabras clave	*Firma, Privacidad, Documentos*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

Los servicios de firma digital son cada vez más usados debido a las necesidades de seguridad y el crecimiento del entorno digital. El reglamento eIDAS establece los requisitos necesarios para que estos servicios puedan ofrecer firma digital cualificada.

La privacidad de un documento que va a firmarse es importante, y su debería estar siempre bajo el control del usuario.

Este proyecto consiste en la definición de un protocolo de firma digital cualificada en la nube en el cual los documentos a firmar no abandonan el dispositivo del usuario.

Abstract (in English, 250 words or less):

Digital signature services are increasingly used due to security needs and the growth of the digital media. eIDAS regulation establishes the necessary requirements with which digital signature services can offer qualified digital signatures.

The privacy of a document that is going to be signed is important and should be under the complete control of the user.

The main purpose of this project consists in the definition of a cloud qualified digital signature protocol in which the documents that are going to be signed do not leave the user's device.

Índice

1. INTRODUCCIÓN 1

1.1. CONTEXTO Y JUSTIFICACIÓN 1
1.2. OBJETIVOS 3
1.3. ENFOQUE Y MÉTODO SEGUIDO 3
1.4. PLANIFICACIÓN 4
1.4.1. Recursos 4
1.4.2. Tareas 4
1.4.3. Diagrama de Gantt 6
1.5. BREVE SUMARIO DE PRODUCTOS OBTENIDOS 7
1.6. ESTADO DEL ARTE 7
1.7. DESCRIPCIÓN DEL RESTO DE CAPÍTULOS. 9

2. ESTUDIO Y DEFINICIÓN DE REQUISITOS 11

2.1. PAdES 11
2.2. CAdES 13
2.3. SAM Y HSM 14
2.4. SERVIDOR DE AUTORIZACIÓN 15
2.5. AUTORIDAD DE CERTIFICACIÓN 16
2.6. BASE DE DATOS 16
2.7. SERVIDOR DE FIRMA 17

3. DISEÑO Y DEFINICIÓN DEL PROTOCOLO 18

3.1. ARQUITECTURA 18
3.2. CASOS DE USO 20
3.2.1. Firma 22
3.2.2. Generación de clave de firma 24
3.2.3. Borrado de clave de firma 26
3.2.4. Algoritmo de generación de la firma digital 27
3.2.5. Proceso de obtención de la firma digital 29

4. IMPLEMENTACIÓN DEL PROTOCOLO 34

4.1. CLIENTE 34
4.2. SERVIDOR DE FIRMA 34
4.3. AUTORIDAD DE CERTIFICACIÓN 34
4.4. SERVIDOR DE AUTORIZACIÓN 34
4.5. SAM Y HSM 34
4.6. CAPTURAS DEL DEMOSTRADOR 35

5. CONCLUSIONES 41

5.1. OBJETIVOS 41
5.2. PLANIFICACIÓN Y METODOLOGÍA 42
5.3. TRABAJO FUTURO 42

6. GLOSARIO 43

7. BIBLIOGRAFÍA 45

Lista de ilustraciones

Ilustración 1: Diagrama de Gantt sobre la organización de las tareas **6**
Ilustración 2: Contenido cubierto en una firma PDF **12**
Ilustración 3: Diagrama sobre la arquitectura del protocolo **18**
Ilustración 4: Diagrama autenticación **20**
Ilustración 5: Diagrama firma digital **22**
Ilustración 6: Diagrama registro identidad **24**
Ilustración 7: Diagrama eliminación identidad **26**
Ilustración 8: Diagrama calculo de firma **29**
Ilustración 9: Creación de identidades **35**
Ilustración 10: Autorización para el registro..... **36**
Ilustración 11: Nueva identidad creada **36**
Ilustración 12: Inicio de un proceso de firma digital..... **37**
Ilustración 13: Documento a firmar cargado en el cliente..... **38**
Ilustración 14: Autorización Firma **39**
Ilustración 15: Construcción documento en el cliente **39**
Ilustración 16: PDF firmado **40**

1. Introducción

1.1. Contexto y justificación

La seguridad de la información en el mundo digital ha ido adquiriendo más importancia a medida que el uso de recursos digitales ha ido incrementando año tras año. Tareas o acciones que antes solo podían realizarse físicamente, han evolucionado y ahora cuentan con la opción digital (enviar correos, chatear, video conferencias…), la firma de documentos no es una excepción, existe la firma digital, y su uso se ha incrementado durante los últimos años.

La firma de documentos tiene una finalidad legal en la que se representa la veracidad de unos datos por parte de una entidad (la que realiza la firma), es por eso, que incluso antes de existir digitalmente, la firma de documentos ha necesitado del cumplimiento de ciertas características de seguridad, como la integridad, autenticidad, privacidad... Este proyecto se centrará en fortalecer la privacidad de los documentos cuando estos son firmados digitalmente.

En el proceso de una firma física, la privacidad del documento firmado, debido a que el contenido de este solo puede ser consultado físicamente, está relacionada con su custodia física, en cambio en un proceso de firma digital no siempre es posible preservar al máximo la privacidad del documento, ya que. al es necesario el uso de un dispositivo y/o servicio de confianza, capaz de generar firmas digitales y con ciertas garantías de seguridad en cuanto a la generación de la firma.

Existen diferentes soluciones de firma digital, este proyecto se centrará en aquellas soluciones que ofrecen un servicio de firma digital en la nube, ya que es en estos casos cuando un documento, para poder firmarse, debe subirse a un servidor, por lo que la privacidad ya no solo depende del usuario.

Una vez dicho lo anterior, en este proyecto se realizará el estudio y diseño de un protocolo de firma digital en la nube, donde el documento que va a ser

firmado no abandonará el dispositivo desde donde se realiza la firma, es decir, que el control sobre el documento seguirá perteneciendo al usuario.

En cuanto a las firmas digitales, en Europa existe el reglamento eIDAS[1], en el cual se establecen unas bases en cuanto a la identificación digital y los servicios digitales de confianza entre los cuales se encuentra la firma digital.

En el reglamento eIDAS[1] pueden encontrarse, entre otros, los requisitos de seguridad que debe un cumplir los proveedores de firma digital para ofrecer QES (Qualified Electronic Signatures).

En lo referido a firma digital cualificada, existen diferentes alternativas, como las tarjetas inteligentes, como el DNI electrónico [12], donde todo el proceso de firma se realiza en local, por lo tanto, existe una privacidad total del documento, ya que las claves de firma se encuentran dentro de las tarjetas, y estas son custodiadas por sus propietarios. Pero para este proyecto se trata de ofrecer una solución donde las claves de firma son custodiadas por un servidor, donde el esquema común en estos casos es subir el documento al servidor donde va a realizarse la firma, y por lo tanto el control de la privacidad del documento ya no solo depende del usuario.

Una vez dicho lo anterior, para poder ofrecer firma cualificada en la nube, en un servicio donde las claves de firma son custodiadas por un servidor, es necesario garantizar que estas solo podrán ser usadas por sus propietarios, por este motivo es necesario un mecanismo o componente que garantice que esto se cumpla, es aquí donde entra en juego el Signature Activation Module (SAM), un componente que cuenta con una certificación Common Criteria[11] EAL4+, bajo el perfil de protección EN-419241-2[10], en el cual se establecen los requerimientos de seguridad que deben cumplirse para ofrecer firma digital cualificada en la nube.

1.2. Objetivos

- Estudio de las necesidades y requisitos de un entorno que preserve la privacidad de los documentos que van a firmarse en un modelo de firma digital en la nube cumpliendo con el reglamento eIDAS[1].

- Diseño de un protocolo de firma digital en la nube en el que se defina un entorno seguro que preserve la privacidad de los documentos que van a firmarse.

- Implementación de un demostrador del protocolo de firma digital en el que se preserva la privacidad del documento que va a firmarse (demostrador).

1.3. Enfoque y método seguido

Con el desarrollo de este proyecto se espera innovar en el ámbito de los servicios de firma digital, actualmente existen soluciones de firma digital en la nube, pero todas ellas pasan por subir el documento que va a firmarse al servidor de firma. Por lo tanto, la estrategia elegida en este proyecto es la de adaptar un protocolo de firma digital en la nube común, es decir aquel en el que se sube el fichero al servidor de firma, a uno en el que el documento que va a firmarse permanece en el dispositivo del usuario, preservando así la privacidad.

El proyecto se desarrollará siguiendo una metodología Agile, aunque en este caso habrá un desarrollador. El proyecto avanzará según iteraciones, donde se realizarán reuniones donde se revisarán los requisitos y resultados obtenidos. El proyecto ha sido dividido en tareas, unas dependientes de otras, lo cual ha sido posible establecer un orden para en cada iteración ir avanzando con el cumplimiento de las tareas.

Dividir el proyecto en tareas es un enfoque adecuado, debido a que así será más fácil medir el progreso a medida que va pasando el tiempo, ya que todas

las tareas constarán de una estimación temporal para asegurar que se llega a cada entrega a tiempo y con las tareas previstas completadas.

1.4. Planificación

1.4.1. Recursos

Para la parte más teórica de este proyecto, se debe conocer cómo deben construirse las firmas digitales en PDF, el funcionamiento y requisitos de los componentes que intervendrán en nuestro protocolo. Por este motivo, el material documental de los estándares (PAdES[2], CAdES[3], CMS[6]...), así como el conocimiento de los protocolos y uso de los componentes que intervienen en el protocolo que va a diseñarse son el recurso principal de este proyecto.

Por otra parte, para la implementación del demostrador del protocolo se dispondrá de conexión a un HSM y del componente SAM (ambos certificados) para la generación de identidades y firma digital cualificada en la nube. También se usará el producto TrustedX, como servidor de autorización que también servirá para obtener las estructuras SAD y SKAD que necesita el SAM para permitir el uso de las claves de los usuarios

1.4.2. Tareas

- **Entrega 1 [14 días]**
    - Plan de trabajo **[7 días]**

- **Entrega 2 [28 días]**
    - Estudio necesidades y requisitos del protocolo **[7 días]**
    - Definición del protocolo. **[5 días]**
    - Diseño y especificación servidor de firma. **[7 días]**

    - Implementación servidor de firma digital (incluye gestión de componentes SAM y DB). **[9 días]**

- **Entrega 3 [28 días]**
    - Estudio necesidades y requisitos entorno para el uso del demostrador. **[5 días]**
    - Diseño y especificación cliente. **[4 días]**
    - Implementación y preparación entorno (AS, CA…). **[6 días]**
    - Implementación cliente y cálculo de firma sobre un documento PDF (Single Page Application, PAdES, …). **[7 días]**
    - Entorno funcional + pruebas. **[6 días]**

- **Entrega Final [35 días]**
    - Memoria. **[35 días]**

- **Presentación en vídeo [7 días]**
    - Preparación y grabación del vídeo. **[7 días]**

- **Defensa [9 días]**
    - Preparación defensa. **[9 días]**

1.4.3. Diagrama de Gantt

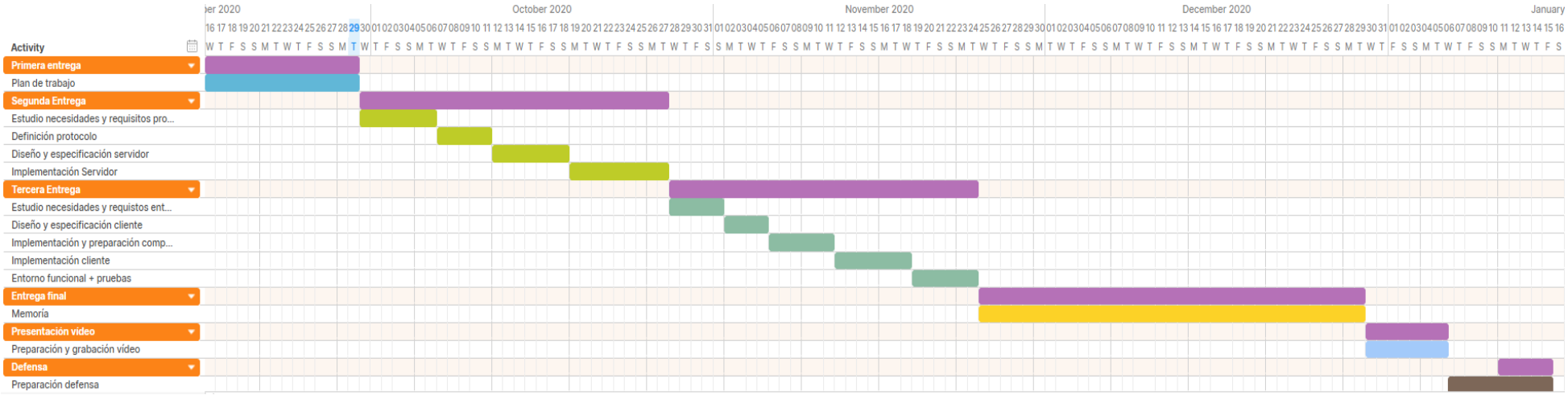

Ilustración 1: Diagrama de Gantt sobre la organización de las tareas

Además, se han ido realizando reuniones con el director, para hacer una revisión del estado de las tareas, analizar si se están cumpliendo las estimaciones y detectar posibles contratiempos en el desarrollo del proyecto.

1.5. Breve sumario de productos obtenidos

Tras finalizar este proyecto se espera obtener como resultado un protocolo de firma digital en el cual la privacidad de un documento que va a ser firmado solo dependa del usuario, es decir, el documento no abandona en claro el dispositivo desde donde se ha iniciado la firma.

Además del protocolo creado también se implementará un demostrador de este centrado en el cálculo de firma digital sobre un documento PDF.

1.6. Estado del arte

En criptografía de clave pública, una firma digital consiste en realizar una serie de operaciones criptográficas sobre unos datos, usando una de las claves de un par de a través de las cuales obtendremos otro conjunto de datos que representan la firma digital, que puede ser verificada con la otra clave del par de claves.

Es necesario establecer una manera común de calcular una firma sobre un documento, ya que de no ser así cada usuario y/o entidad construirían y realizarían la firma de forma diferente. Los estándares de firma digital se encargan establecen como debe ser una firma digital, y que información debe incluir esta. Algunos ejemplos de estándares de firma establecidos por eIDAS[1] como formatos de firma avanzada son CAdES[3] (firma de binarios), XAdES (firma de documentos XML), o PAdES[2] (firma de documentos PDF). En los estándares se definen diferentes perfiles de firma en función de la información añadida a la firma, por ejemplo, sellos de tiempo, material de validación verificación, activación de la firma… Por ejemplo, algunos de los perfiles de firma PAdES son los siguientes:

- PAdES-BES: Firma básica que incluye los campos necesarios para la verificación de la firma.

- PAdES-EPES: debe incluir la misma información que se incluiría en una firma PAdES-BES, pero además debe incluir una política de firma.

- PAdES-T: Se construye,sobre una firma PAdES-BES o PAdES-EPES, y además debe incluir un sello de tiempo.

Una de las ventajes de usar alguno de estos estándares establecidos por eIDAS(CAdES[3], XAdES, PAdES[2]) es que al estar establecido en toda Europa, permite la interoperabilidad entre todos los estados miembro.

El reglamento eIDAS[1] establece, entre otros, los requisitos que deben cumplir en los servicios de firma digital, en concreto, en el estándar Common Criteria[11], se encuentra el perfil (EN-419241-1), en el cual se explica en detalle los requerimientos de seguridad para la firma digital con identidades en la nube. La segunda parte de este documento (EN-419241-2[10]) define los requerimientos de seguridad que deben cumplir los dispositivos que generan firmas digitales (Qualified Electronic Signature Creaction Device – QSCD), para que estas sean cualificadas (Qualified Electronic Signatures - QES), por ejemplo, una clave de usuario generada y custodiada en un servidor solo debe poder ser usada por este.

En un entorno de firma cualificada debe existir un dispositivo certificado encargado de realizar las firmas digitales, en el caso de este proyecto, se trata del Hardware Security Module (HSM), un hardware especializado en operaciones criptográficas, que servirá para la generación de claves y firmas digitales.

El componente encargado de hacer uso de las operaciones ofrecidas por un HSM es el Signature Activation Module (SAM), se trata de un componente certificado Common Criteria[11] con el perfil EN-419241-2[10] que se ha mencionado anteriormente, encargado de hacer uso de las operaciones ofrecidas por un HSM garantizando el cumplimiento de la normativa eIDAS[1] para la protección de claves y firma digital cualificada.

Este proyecto se centrará en el estudio y diseño de una solución de firma digital cualificada en la nube que cumpla con el reglamento eIDAS[1], manteniendo la privacidad de los documentos que van a firmarse, por lo tanto en el diseño del protocolo se tendrá en cuenta la presencia del componente SAM como proveedor de firma (y por lo tanto se estará usando también un HSM).

Uno de los requisitos indispensables para poder ofrecer un servicio de firma digital en la nube donde las claves de firma son custodiadas en el servidor, es el de garantizar la autenticidad de la identidad de los usuarios, es por eso que deben establecerse los mecanismos oportunos de autenticación según lo establecido en eIDAS[1], y es también por este motivo que los servidores de autorización cumplen un papel importante dentro de un entorno en el que quiere ofrecerse firma digital cualificada en la nube.

En lo referido a la autenticación de los usuarios, hay que destacar que en un entorno de firma digital cualificada, tal y como está establecido en el reglamento eIDAS[1], los usuarios son autenticados por medio de un servidor de autorización mediante mecanismos de autenticación considerados fuertes, por ejemplo biometría o la presencia de un token físico, pero además de esta protección, el componente SAM también protege las claves de los usuarios, de esta manera, para garantizar la clave que va a usarse pertence al usuario, se establecen como mínimo 2 factores de autenticación: uno o más aportados por el servidor de autorización y otro por parte del componente SAM.

1.7. Descripción del resto de capítulos.

Los siguientes capítulos de este proyecto se desarrollarán en función de los objetivos definidos, es decir:

- En el capítulo 2 se hará un estudio sobre los diferentes componentes y estándares que intervendrán en el protocolo.

- En el capítulo 3 se definirá el protocolo, como ya se habrá visto el papel de todos los componentes en el capítulo anterior será más fácil entender la solución propuesta.

- En el capítulo 4, como ya se habrá definido el protocolo, se explicará cómo se ha implementado un demostrador de este.

2. Estudio y definición de requisitos

Para poder diseñar el protocolo de firma digital en la nube de este proyecto, antes hay que conocer los componentes y elementos que intervendrán o serán creados durante un proceso de firma digital.

2.1. PAdES

Un tipo de documentos muy extendido en el mundo digital consiste en los documentos PDF. Un documento PDF, se trata de unos datos organizados siguiendo un formato (Portable Document Format – PDF 1.7 – ISO 32000 - 1 [5]) de manera que un lector de documentos en formato PDF puedo enseñarlos (y también operar con ellos), en función de la información contenida en los mismos.

En ISO 32000 – 1 [5], podemos encontrar la definición del formato PDF, uno de los contenidos de este documento (en concreto el apartado 12.8), está dedicado a la firma digital, y como debe ser esta en un documento PDF.

PAdES (PDF Advanced Electronic Signatures) [2] [4], se trata de un estándar que complementa el formato de las firmas digitales PDF, a partir de una definición más específica, así como la adición/modificación de elementos del formato, con la finalidad de mejorar la seguridad de estas, así como añadir características de interoperabilidad, longevidad y validación.

Existen diferentes niveles/tipos de firma PAdES[2], cada uno de ellos incluye ciertos atributos característicos. Este proyecto se centrará en el formato PAdES-BES[4].

Es necesario destacar, que la parte correspondiente a la firma de un PDF consiste en la inclusión de una firma CAdES (CMS Advanced Electronic Signature)[3], dentro del documento PDF.

Dentro de un documento PDF, la firma se encuentra dentro de la sección “Signature Dictionary”, dentro del cual se encuentra el campo “ByteRange”, el cual indica el contenido cubierto por la firma. La firma debe cubrir todo el documento exceptuándose a sí misma. La firma se guarda en el campo “Contents” dentro de la sección “Signature Dictionary”.

Tal y como se indica en la especificación PADES-BES [4], la firma CAdES debe ser computada sobre todo el documento a excepción de la firma.

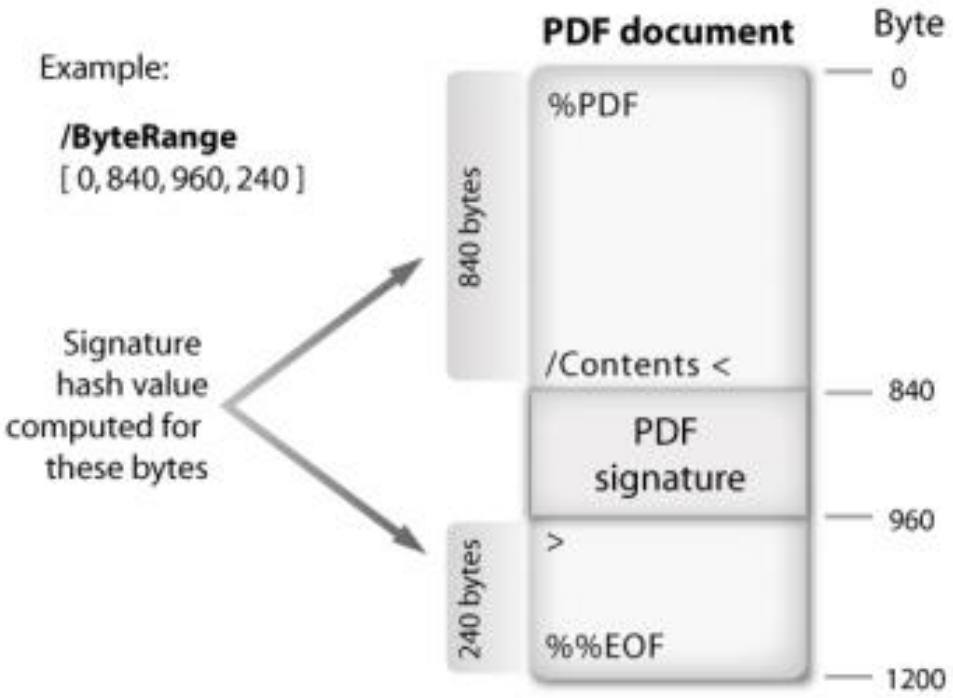

Ilustración 2: Contenido cubierto en una firma PDF

El valor del campo “SubFilter” de la sección “Signature Dictionary” del PDF debe contener el valor “ETSI.CAdES.detached”, este campo describe el tipo de contenido que se guarda en el campo “Contents”, que en este caso corresponde con una firma CAdES-BES y es una firma de tipo detached ya que el CMS de la firma no incluye el documento que se está firmando.

No se debe usar el campo “Cert” de “Signature Dictionary”.

2.2. CAdES

CAdES[3] se trata de un estándar para la firma de binarios, es decir, cualquier tipo de contenido digital, ya que en el mundo digital todo está compuesto por unos y ceros.
Al igual que con PAdES[2], CAdES[3] presenta una serie de directrices para realizar uno firma sobre unos datos. El estándar PAdES[2] se construye sobre PDF[5], de manera similar, el estándar CAdES[3] se construye sobre CMS[6] basado en la sintaxis PKCS#7.

Al igual que las firmas PAdES[2], las firmas CAdES[3] pueden ser de diferentes tipos dependiendo de la información que incluyan y de las características que debe tener la firma. Este proyecto se centrará en el perfil CAdES-BES.

Una firma CAdES[3], contendrá el campo “SignerAttributes” en el cual se irán todos aquellos atributos que deben ir cubiertos por la firma digital. La especificación CMS de CAdES[7], establece cuales son los atributos que debe contener como mínimo una firma, para ser considerada CAdES-BES:

- Message-digest: Hash del contenido que quiere firmarse. En el caso de este proyecto el valor de este campo se corresponde con el hash del documento PDF especificado por el campo “ByteRange”.
- Content-type: Indica el tipo de contenido que va a firmarse.
- ESS signing-certificate o ESS signing-certificate-v2, referencia al certificado del firmante (que se encuentra en el campo “Certificates” de la firma.

Además de estos atributos obligatorios la firma CAdES[3] debe contener la estructura e información mínimas establecidas por la especificación CMS[6].

La especificación PAdES-BES [4], indica que la firma CAdES:

- Debe incluir un único elemento “SignerInfo”, y se indica que la construcción del contenido de la firma CAdES, ha de realizarse según lo definido en la especificación CAdES [3] y CMS [6].

- El campo “content-type” de la firma CAdES debe contener el valor “id-data”.

- El campo message-digest debe ser definido como se indica en CMS [5].

- La firma CAdES debe incluir una referencia un certificado en el campo “signing-certificate” o “signing-certificate-v2”, que esté presente en el campo “certificates” de la firma (cabe recordar que en el PDF el campo cert no debe usarse).

En cuanto a la generación del hash que usará para la firma digital, la especificación CMS[6], explica que en el caso de existir “Signed Attributes” el hash resultante debe ser el resultado de realizar el hash sobre todos estos atributos concatenados en orden de aparición en la firma y en codificación DER. Es necesario destacar que como los “Signed Attributes” incluyen los campos “Content-type” y “Message-digest” el contenido que va a firmarse, en este caso el documento PDF, queda cubierto gracias a la presencia de estos campos, ya que al firmar el campo “Message-digest”, el cual incluye el hash del documento, se está firmando indirectamente la integridad del documento.

2.3. SAM y HSM

El Hardware Security Module (HSM), se trata de un dispositivo capaz de ejecutar operaciones criptográficas como la creación de claves, hashes, firmas digitales… Se trata de un componente certificado, que tiene un papel importante en cuanto a firma digital cualificada, ya que dispone de medidas de seguridad (por ejemplo, anti-tampering), y ofrece mecanismos de protección de claves y auditoria.

El Signature Activation Module (SAM), es un componente que usa el HSM para ofrecer un servicio de generación de claves y firma digital. Es importante destacar que se trata de un componente certificado Common Criteria EAL4+ bajo el perfil EN-419241-2[10]. Un componente que es certificado como Common Criteraia[11], implica un reconocimiento y garantía de las medidas de seguridad con las que este cuenta y con las que ha sido desarrollado.

El conjunto formado por SAM y HSM son los componentes que hacen posible ofrecer un mecanismo de protección de claves que garantiza solo los propietarios de las claves custodiadas por el servidor de firma pueden hacer uso de estas.

Para poder hacer uso de las operaciones del SAM, es necesario disponer de un servidor de autorización capaz de generar los privilegios exigidos por SAM en cada una de las operaciones que necesitan de la autorización del propietario de la clave que se está usando. Estos privilegios son:

- Signature Activation Data (SAD): Estructura necesaria cuando va a realizarse una firma, que viene firmada por el servidor de autorización y es comprobada por el componente SAM.

- Signing Key Activation Data (SKAD): Estructura necesaria cuando va a realizarse alguna operación con el componente SAM diferente de la firma, que viene firmada por el servidor de autorización y es comprobada por el componente SAM.

2.4. Servidor de Autorización

Con tal autenticar y autorizar a los usuarios será necesario el uso de un servidor de autorización. Las correspondientes autorizaciones deben poder realizarse mediante el protocolo OAuth 2.0 [8][9], mediante el uso de un grant

de tipo “Authorization Code”, es decir obtención de un token mediante el intercambio de un code por un token.

En cuanto a la autenticación del cliente, el servidor de autorización también deberá soportar el grant “Client Credentials”, es decir, que el cliente se autentica mediante un secreto.

Otro detalle que debe tenerse en cuenta es que el servidor de autorización tiene que ser capaz de generar las estructuras SAD y SKAD (explicadas en el apartado anterior), de manera que el servidor de firma pueda hacer uso del SAM, ya que estas estructuras son la evidencia de que el usuario quiere hacer uso de sus claves (solo él puede hacerlo).

2.5. Autoridad de Certificación

Para entender la necesidad de incluir una CA (Certification Authority) en el protocolo, hay que recordar que en la firma CAdES[3], con tal de cumplir con el estándar, necesita incluir el certificado del firmemente en el CMS de la firma.

El componente SAM genera un par de claves, pero no el certificado asociado a estas, por lo que es necesaria la creación de este mediante una Autoridad de Certificación (CA). De esta manera se asociada una clave generada y custodiada por el servidor de firma a un usuario.

2.6. Base de datos

Componente necesario para almacenar las claves del usuario, así como la información necesaria que se va recopilando durante el protocolo: identificador, certificado de firma…

2.7. Servidor de firma

El servidor de firma será el encargado de realizar la comunicación con el componente SAM, y por lo tanto será el encargado de construir parte del formato de la firmal, por lo que deberá ser capaz de generar firmas CMS[6]. El servidor debe ser capaz de completar el protocolo OAuth2[8][9] para la obtención de la autorización de los usuarios.

2.8. Cliente de firma

En el protocolo diseñado debe mantener la privacidad del documento, también es conveniente pensar en que el proceso de firma debería ser simple para los usuarios. Por este motivo, un cliente que se ejecuta en el navegador parece conveniente, ya que si el dispositivo desde donde se inicia la firma dispone de un navegador (PC, móvil, SmartTV…) ya sería apto para poder usar el servidor de firma.

El cliente de navegador se comunica con el servidor de firma, y ha de ser capaz de generar parte del formato de la firma, en concreto la parte referida a los campos del PDF, así como la inclusión del CMS en la firma una vez se ha generado la firma.

3. Diseño y definición del protocolo

3.1. Arquitectura

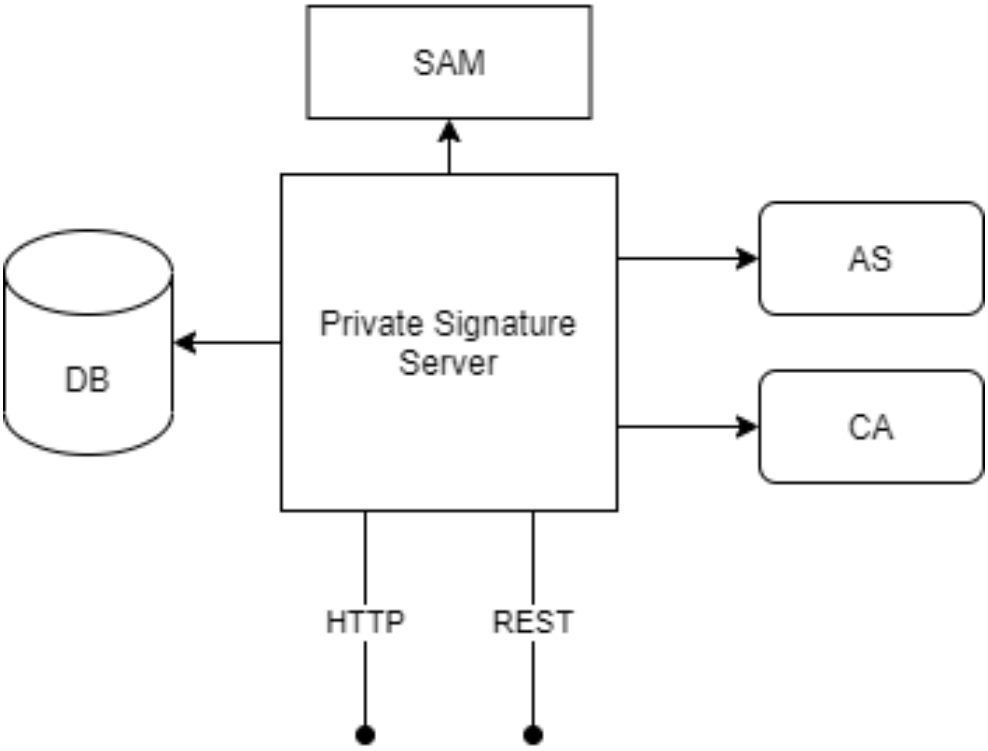

Ilustración 3: Diagrama sobre la arquitectura del protocolo

La imagen anterior es un esquema en el que se representan los componentes que intervienen en el protocolo de firma:

- **Private Signature Server:** Servidor encargado de ofrecer las operaciones de firma y de construir el formato de la firma. El servidor puede dividirse en dos interfases:
    - **REST**: Interfaz en la que estarán disponibles las operaciones necesarias para poder realizar una firma digital, así como de autenticación para poder hacer uso de estas operaciones.

- **HTTP**: Interfaz que se encargará de descargar el cliente un cliente que se comunica con la interfaz REST del servidor, también se encarga de construir parte del formato de la firma.

- **SAM**: Componente que se encarga de las operaciones criptográficas del servicio (generación de claves, firma…), por medio de la comunicación con uno o más dispositivos HSM (Hardware Security Module). Este componente, juntamente con el cumplimiento de ciertos requisitos en el entorno en el que se ejecuta, hacen que las firmas en el servidor cumplan con el reglamento eIDAS[1]. Cuando los usuarios de este servidor generen una clave de firma o realicen una firma digital, el servidor delegará estas tareas al SAM para que este pueda gestionarlas a través del dispositivo HSM.

- **AS**: Servidor de autorización encargado de autorizar el acceso a los servicios ofrecidos por este servidor.

- **CA:** Componente que se encargará de certificar las claves generadas por nuestro servicio, para ello, una vez se genere una clave de firma (par de claves), el servidor iniciará un proceso de certificación de la clave pública del par de claves que ha generado el SAM

- **DB**: Base de datos donde se almacenarán los usuarios que pueden acceder a los servicios del servidor, así como las correspondientes claves de firma generadas e información necesaria para su uso (certificado.

3.2. Casos de uso

A continuación, se explicarán los 3 casos de uso que se trabajarán en este proyecto. Se asumirá que el usuario ya está autenticado y autorizado para usar la correspondiente operación de la aplicación. La siguiente imagen describe el proceso de obtención de autorización para hacer uso de alguna de las operaciones del servidor

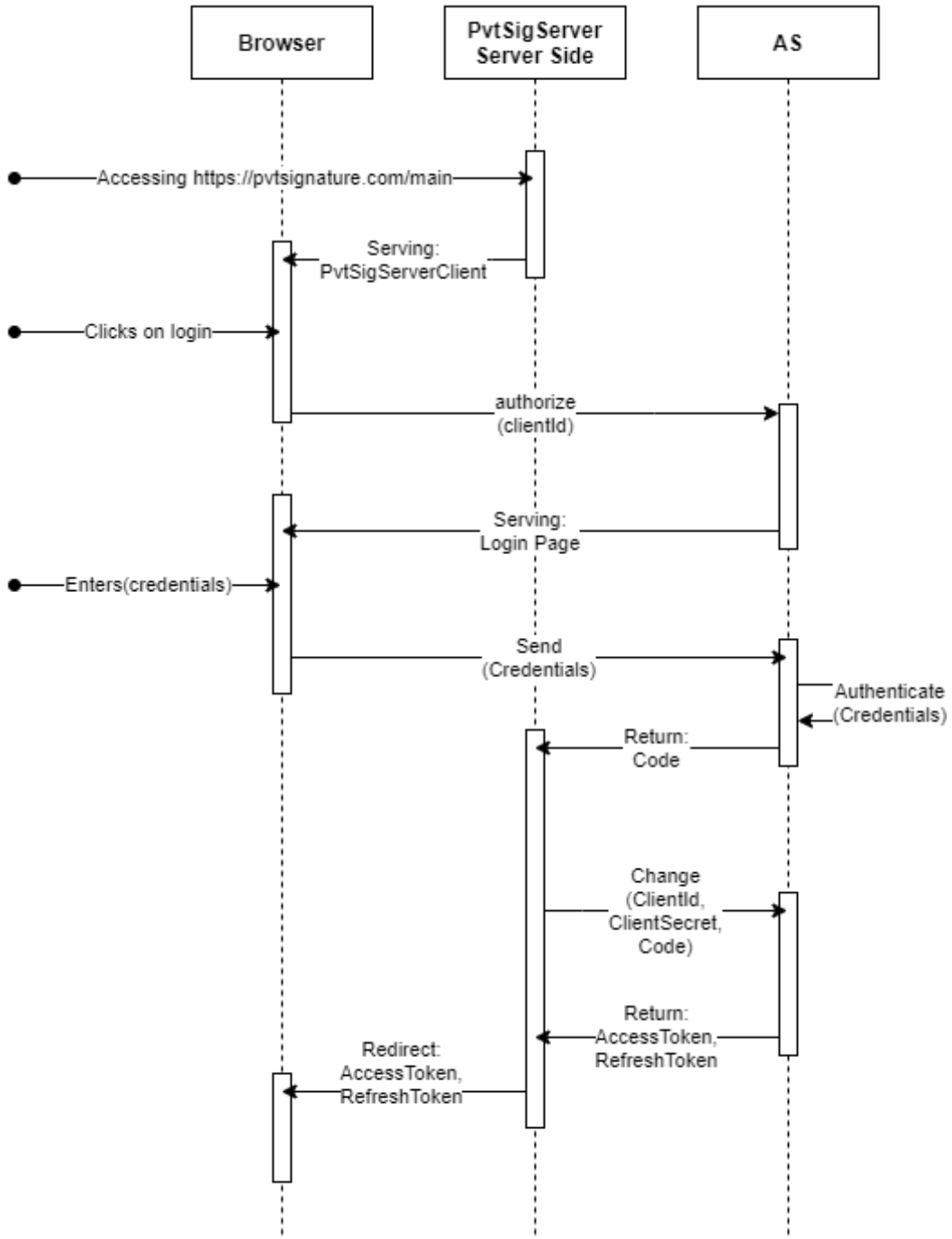

Ilustración 4: Diagrama autenticación

Se ha de tener en cuenta que el cliente es una SPA que se ejecuta en el navegador, por lo tanto, el secreto del cliente se encuentra en el servidor.

Una vez se ha obtenido el correspondiente token de autorización, es enviado a la SPA, para que, mediante su inclusión en posteriores invocaciones al servidor, esta pueda hacer uso de las correspondientes operaciones de este.

Como proteger el token de autorización en la aplicación depende de la implementación. Una manera, consiste en pasar los tokens como cookies y las correspondientes cabeceras https para garantizar la seguridad de acceso a estos.

3.2.1. Firma

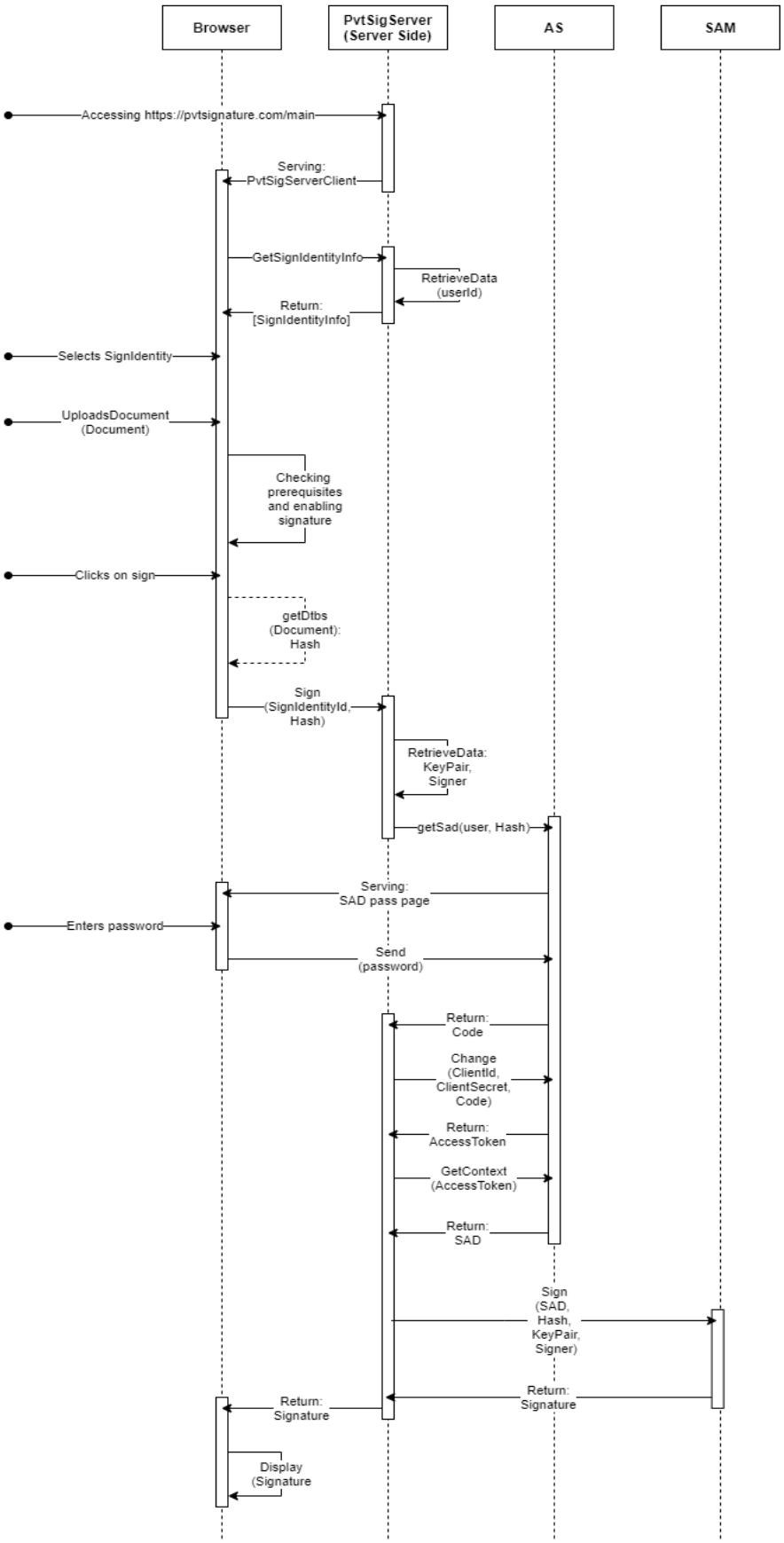

Ilustración 5: Diagrama firma digital

La imagen anterior corresponde al proceso realizado cuando un usuario realiza una operación de firma con el servidor. En el proceso se asume que el usuario ya ha solicitado la autorización al servidor de autorización (tal y como se explica al principio de esta sección), por lo tanto, la SPA ya cuenta con un token de autorización que irá reutilizando a lo largo del proceso.

En el diagrama de firma, la parte correspondiente al cómputo del hash que va a firmarse se enseña en puntos suspensivos ya que este proceso se explicará con más detalle en un apartado posterior.

En el diagrama también puede verse otra petición al servidor de autorización, se trata de una petición para obtener la estructura SAD proporcionada por el servidor de autorización (y firmada por el mismo) en la cual se encuentra el hash de los datos que van a firmarse, la obtención de esta estructura es necesaria, ya que el SAM tiene registrado el servidor de autorización y para firmar unos datos necesita que el servidor de autorización haya autorizado la firma de los datos, ya que este habrá sido invocado por el usuario que quiere realizar la firma.

3.2.2. Generación de clave de firma

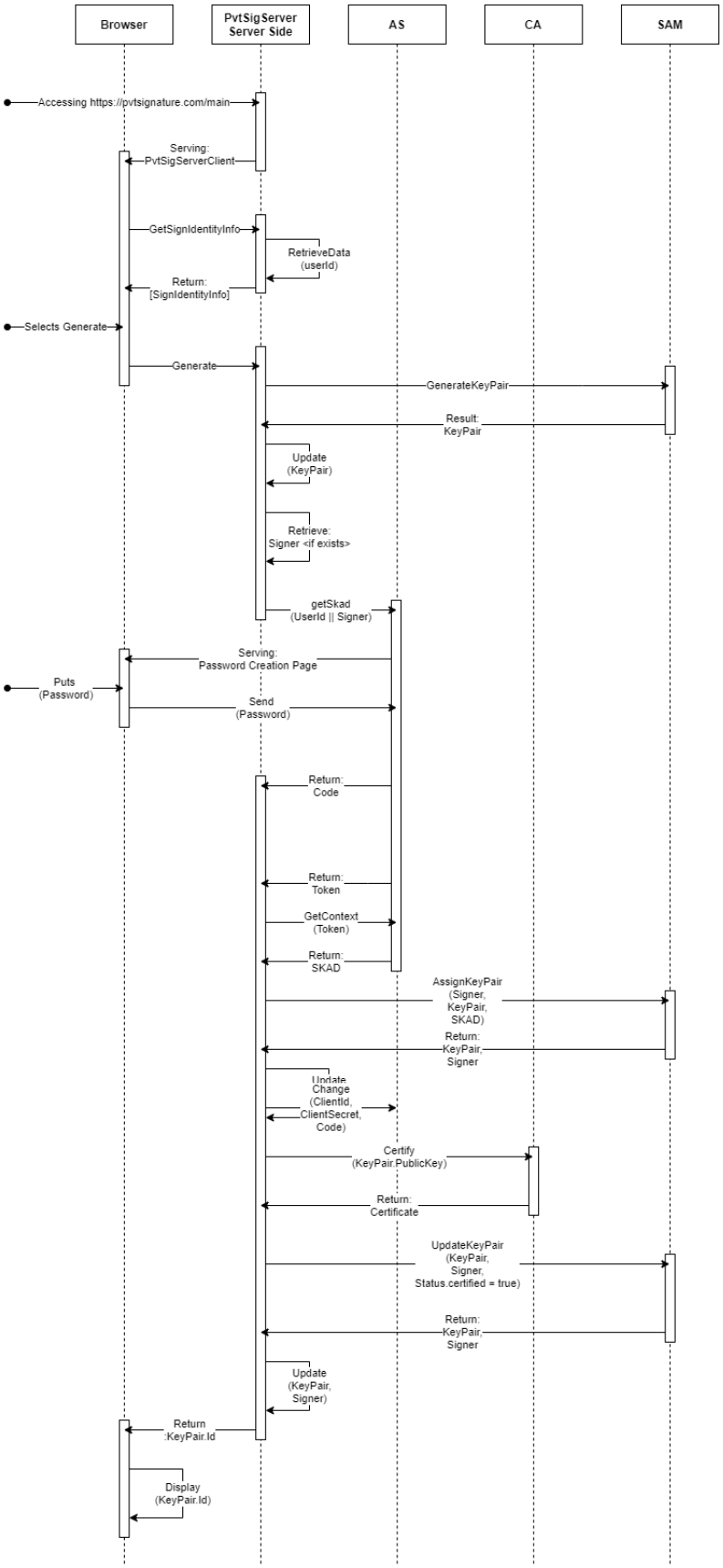

Ilustración 6: Diagrama registro identidad

La imagen anterior corresponde al proceso realizado cuando un usuario solicita la creación de una clave de firma en el servidor. En el proceso se asume que el usuario ya ha solicitado la autorización al servidor de autorización (tal y como se explica al principio de esta sección), por lo tanto, la SPA ya cuenta con un token de autorización que irá reutilizando a lo largo del proceso.

En el diagrama también puede verse otra petición al servidor de autorización, se trata de una petición para obtener la estructura SKAD proporcionada por el servidor de autorización (y firmada por el mismo) en la cual se encuentra, el valor de activación de la clave que es generada por el servidor de autorización tras aplicar N factores de autenticación (contraseña, un token, biometría, etc), la obtención de esta estructura es necesaria, ya que el SAM tiene registrado el servidor de autorización y para firmar unos datos necesita que el servidor de autorización haya autorizado la firma de los datos, ya que este habrá sido invocado por el usuario que quiere realizar el registro de una nueva identidad en el servidor.

3.2.3. Borrado de clave de firma

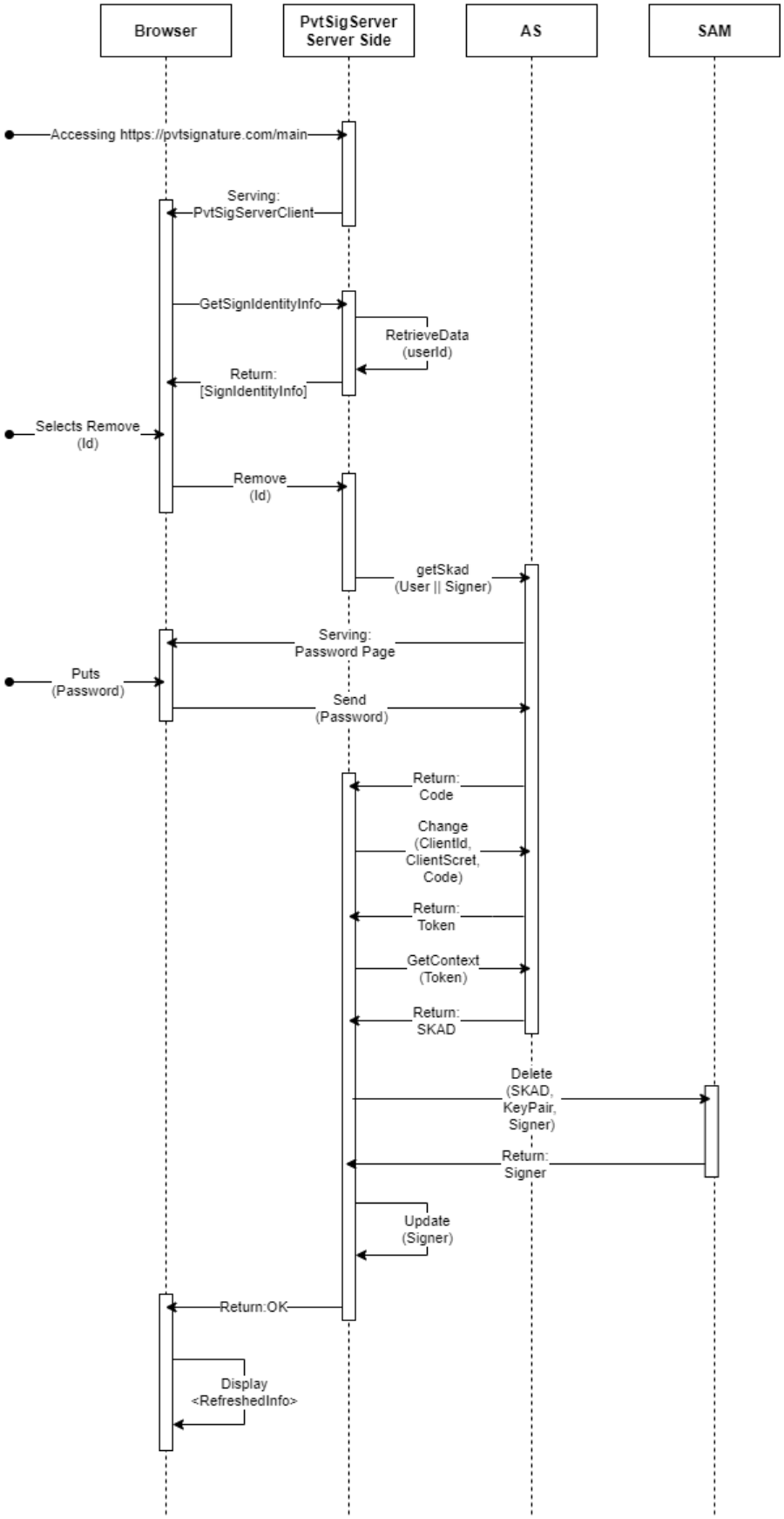

Ilustración 7: Diagrama eliminación identidad

La imagen anterior corresponde al proceso realizado cuando un usuario solicita la eliminación de una clave de firma en el servidor. En el proceso se asume que el usuario ya ha solicitado la autorización al servidor de autorización (tal y como se explica al principio de esta sección), por lo tanto, la SPA ya cuenta con un token de autorización que irá reutilizando a lo largo del proceso.

En el diagrama también puede verse otra petición al servidor de autorización, se trata de una petición para obtener la estructura SKAD proporcionada por el servidor de autorización (y firmada por el mismo) en la cual se encuentra, el valor de activación de la clave que es generada por el servidor de autorización tras aplicar N factores de autenticación (contraseña, un token, biometría, etc), la obtención de esta estructura es necesaria, ya que el SAM tiene registrado el servidor de autorización y para firmar unos datos necesita que el servidor de autorización haya autorizado la firma de los datos, ya que este habrá sido invocado por el usuario que quiere realizar el eliminación de la clave de firma en el servidor.

3.2.4. Algoritmo de generación de la firma digital

El objetivo del diseño de este protocolo consiste en mantener la privacidad del del documento que va a firmarse, por este motivo es conveniente que este no abandone el dispositivo del firmante (ordenador, móvil, Smart TV…). Una vez dicho lo anterior, el proceso de firma, y en concreto la generación de la firma digital dispone de una serie de pasos, algunos de ellos deben ejecutarse en el cliente (navegador) y otros en el servidor de firma. Una vez dicho lo anterior, y antes de explicar el computo de la firma, es necesario mencionar las siguientes aclaraciones:

1. El servidor no puede recibir en ningún caso el documento que va a firmarse “en claro”, pero si puede recibir el hash del documento para montar el CMS correspondiente a la firma. Si el servidor recibe el hash del documento no se incumple con el protocolo, ya que a partir del hash no podemos deducir el contenido del documento.

2. Para poder calcular el hash del documento en el cliente, necesitamos información sobre la firma CAdES que va a generarse, en concreto cuanto espacio ocupará esta, ya que el hash cubre el campo “ByteRange” que define el contenido cubierto por la firma y del cual se puede inferir el tamaño de esta.

3. El tamaño de la firma CAdES puede deducirse con la información disponible en el servidor, ya que en este se encuentra toda la información necesaria para deducir el tamaño, ya sea porque tiene disponibles lo datos necesarios (tamaño del certificado, tamaño campo content-type o signing-time), o porque puede deducirlo a partir de otros campos (el tamaño de la firma y el hash pueden deducirse a partir de los correspondientes algoritmos).

3.2.5. Proceso de obtención de la firma digital

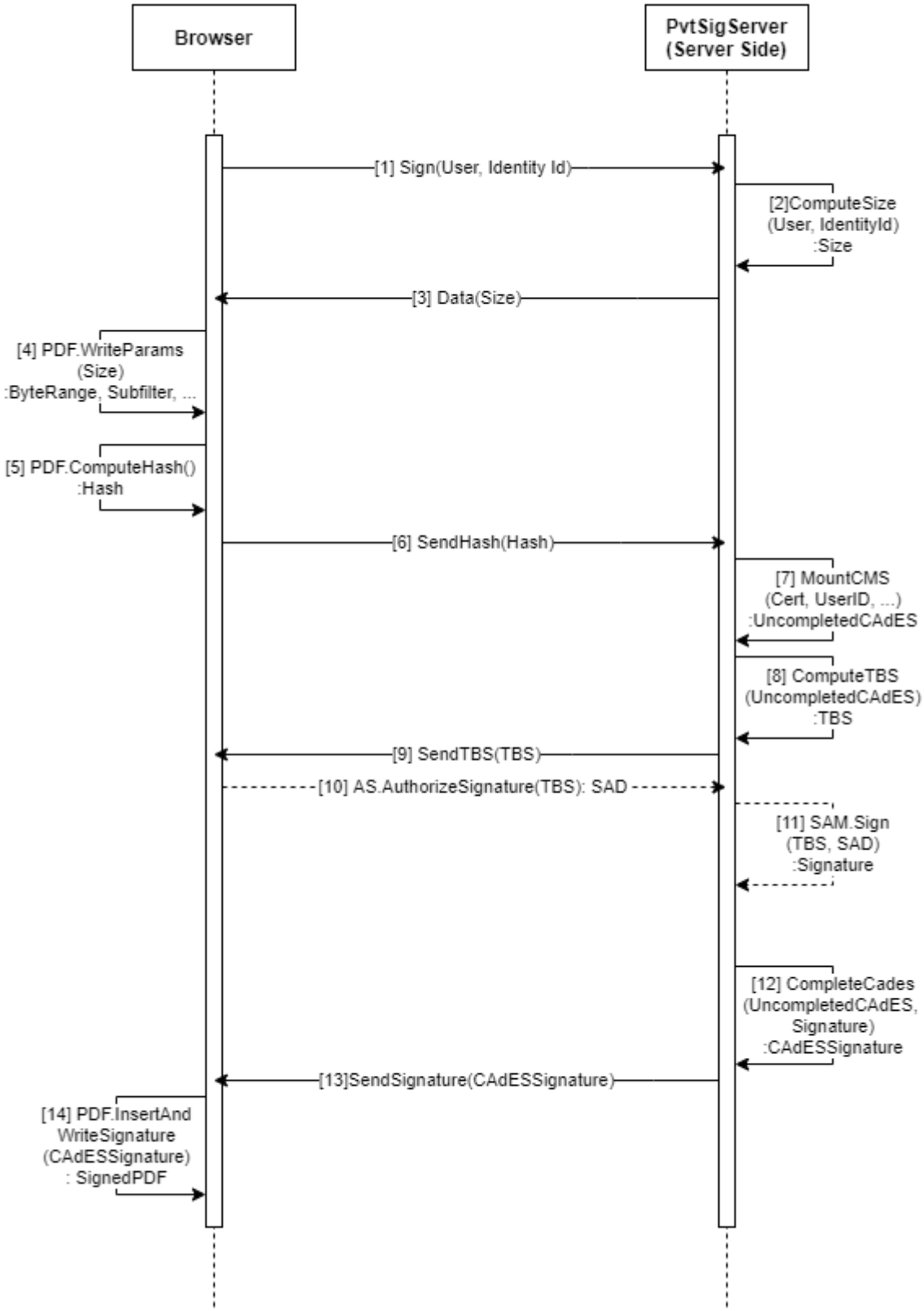

Ilustración 8: Diagrama cálculo de firma

El diagrama anterior describe el proceso de obtención de una firma PAdES-BES. Se asume que el documento ya está cargado en el cliente y que el usuario ya está autenticado y autorizado para realizar una firma digital sobre el documento. Para no complicar el diagrama, solo se han dibujado la parte del navegador y del servidor, ya que solo existen dos pasos que son realizados por componentes que no son estos (SAM y servidor de autorización), pero se han indicado estas acciones con líneas suspensivas para que se tenga en cuenta que estas acciones no son llevadas a cabo directamente por ni por el navegador ni por el servidor.

Antes de empezar con la explicación del algoritmo es necesario aclarar que en todas las peticiones que se realizan al servidor es necesaria la presencia del token de autorización, y que se verifica en cada llamada. Una vez dicho lo anterior, no se incluirán explicaciones sobre la autenticación del usuario durante la explicación del algoritmo del cálculo de la firma.

A continuación, se explicará cada paso del diagrama:

1. El usuario tiene cargado en el cliente el documento, y ha seleccionado el identificador de una de sus identidades, que están custodiadas en el servidor de firma. A continuación, envía una petición al servidor de firma indicando que quiere realizar una firma con la clave de firma que tiene el identificador que ha seleccionado antes.

2. El servidor al recibir la petición inicia un proceso de firma., por lo que recupera la identidad de firma, el certificado y más información de esta de la base de datos a partir de la cual puede estimar el tamaño que tendrá la firma CAdES[3] generada, esto es posible que en el servidor esta disponible toda la información necesaria para realizar esta estimación (certificado, id del usuario…), e incluso puede inferir el tamaño de la información que no dispone gracias al resto de información (el hash con el algoritmo de hash, y la firma

con el algoritmo de firma). El tamaño estimado de la firma podría ser calculado cuando se realiza el registro de la identidad en el servidor y guardar este valor en la base de datos

3. Una vez el servidor a estimado el tamaño que tendrá la firma, devuelve este valor al cliente, ya que este lo necesita para reservar espacio en el PDF para la firma según lo establecido en PAdES[2].

4. El cliente recibe como respuesta del servidor, el tamaño que ocupará la firma digital, y empieza a modificar el PDF, añadiendo los campos correspondientes según el estándar PAdES[2] (en el caso de este proyecto para conseguir una firma PAdES-BES[4]): Se añade el valor "ETSI.CAdES.detached" al campo "subfilter" de la sección "Signature Dictionary", se establece el contenido que será cubierto por la firma añadiendo el campo "Byte-range" con los valores correspondientes a partir del tamaño de firma obtenido por el servidor, y se añade el campo "Contents" del tamaño que tendrá la firma y se rellena con zeros.

5. El cliente a partir del "ByteRange" que ha definido calcula el hash del documento.

6. Una vez se ha calculado el hash del paso anterior, el cliente envía este hash al servidor.

7. El servidor recibe el hash del documento y empieza a construir el CMS para la firma según el estándar CAdES[3] (en el caso de este proyecto para construir un perfil CAdES-BES), se añade el certificado del firmante al CMS, así como la correspondiente referencia dentro de los "signed atributes" del CMS, también se añade el hash como "Signed Attribute" en concreto será el atributo "Message Digest", además se aprovecha para añadir otros atributos

firmados (como “Content-type”). En este paso el CMS aún no está completo.

8. Una vez se ha construido parte del CMS, el servidor calcula el hash del CMS tal y como se especifica en el estándar CAdES – CMS [7], es decir se calcula un hash de los “Signed Attributes” concatenados y codificados en DER.

9. El servidor una vez ha calculado el hash que va a firmarse en el paso anterior, se lo devuelve al cliente.

10. El cliente inicia un flujo de autorización con el servidor de autorización, donde el usuario deberá introducir sus credenciales y el secreto que protege su clave. El resultado de este flujo de autorización es que el servidor ya dispondrá de la estructura SAD firmada por el servidor de autorización que necesitará el componente SAM para poder firmar los datos.

11. El servidor de firma envía el hash juntamente con la estructura SAD al componente SAM, para que este realice la firma. El componente SAM realizará la firma y la devolverá al servidor de firma.

12. Una vez el servidor de firma recibe la firma que ha calculado el componente SAM, se añade la firma al CMS que se empezó a construir en el paso 7, una vez se ha añadido la firma al CMS este ya cuenta con todos los atributos necesarios para ser considerado una firma CAdES-BES según lo establecido en el estándar CAdES[3].

13. El servidor devuelve como respuesta la firma CAdES[3] al cliente.

14. El cliente al recibir la firma CAdES[3], la inserta en la posición correspondiente del PDF (a partir del “ByteRange”), completando el formato y obteniendo así un PDF con una firma PAdES[2].

4. Implementación del protocolo

4.1. Cliente

El cliente ha sido implementado con ReactJS y Typescript, ya que debía poderse ejecutar en el navegador era necesario que este sea implementado en JavaScript, o un framework del mismo.

Se ha decido usar ReactJS por que permite implementar de manera sencilla una Single Page Application (SPA), y que resultará útil a la hora de realizar operaciones en una misma pantalla, así como mantener el estado de la misma.

4.2. Servidor de firma

El servidor ha sido implementado en Golang, se ha usado este lenguaje por la sencillez que aporta a la hora de implementar un servidor, como la definición sencilla de objetos JSON o asn1, cliente REST…

4.3. Autoridad de certificación

Al igual que con el servidor de firma, se ha implementado este componente en Golang.

4.4. Servidor de autorización

Se ha usado el producto TrustedX de Entrust como servidor de autorización, ya que este soporta la generación de los privilegios que necesita el componente SAM, y permite establecer mecanismos de autenticación fuertes para los usuarios.

4.5. SAM y HSM

Se ha usado el componente SAM de Entrust que dispone de la certificación Common Criteria[11] EAL4+ bajo el perfil EN-419241-2[10], el cual hace uso de un HSM nCipher de Entrust.

4.6. Capturas del demostrador

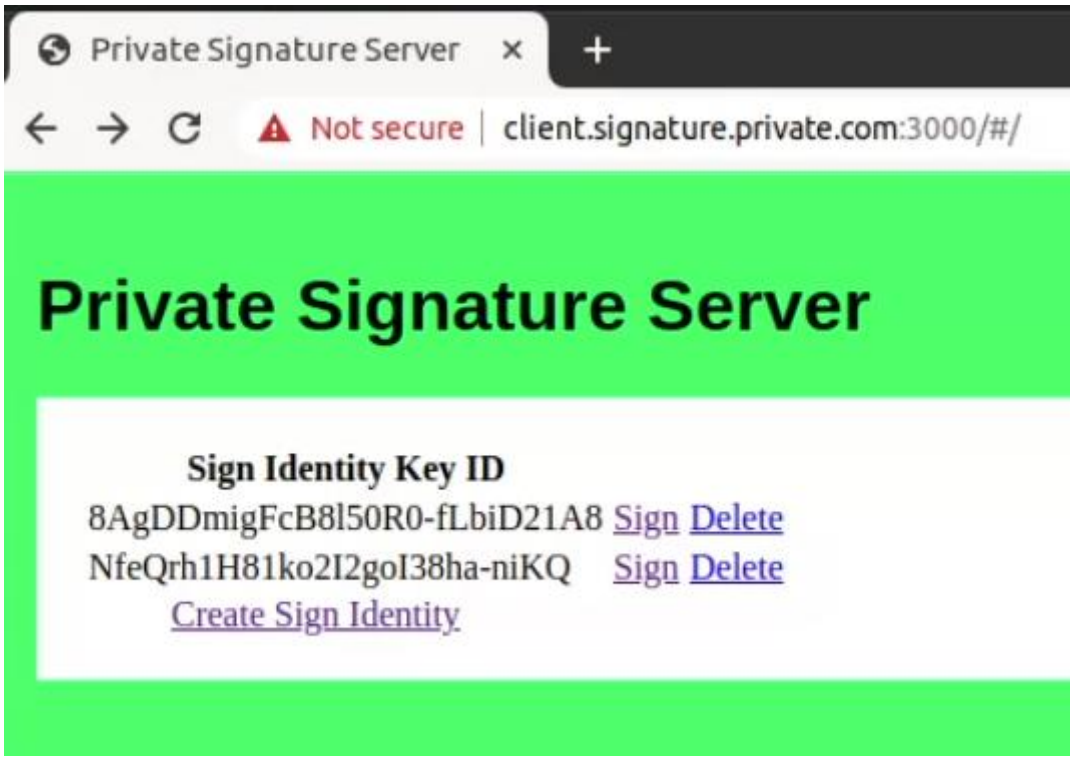

Ilustración 9: Creación de identidades

En la imagen anterior puede verse la pantalla de inicio del cliente, como se ha comentado anteriormente se trata de una aplicación web, en concreto una Single Page Application (puede notarse que el cliente esta ejecutandose en un navegador.

Tambien, en el caso concreto de la imagen, puede notarse que que ya existen dos identidades de firma custodiadas por el servidor de firma, y que por cada una de ellas tenemos dos opciones:

- Borrar: En cuyo caso se eliminaría la identidad del servidor.
- Firmar: En cuyo caso de firmaría un documento. Este proceso se verá con más detenimiento más adelante.

Tambien puede verse la opción de crear otra identidad, si se selecciona esta opción el cliente de navegador redirecciona al usuario hacia el servidor de

autorización, y se completa un flujo de autenticación como el visto en la definición del protocolo:

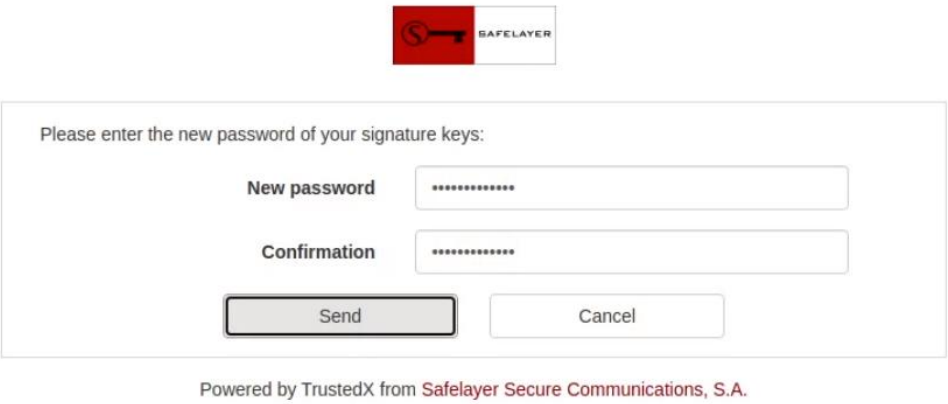

Ilustración 10: Autorización para el registro

En la imagen anterior, puede apreciarse como el servidor de autorización, después de solicitar las credenciales del usuario, solicita un secreto para la protección de la clave.

Después de completarse el flujo de autorización, el servidor de autorización habrá generado la estructura SKAD, que es recuperada por el servidor de firma con el token con ayuda del token de autorización, que se usa para generar la identidad con el componente SAM.

Ilustración 11: Nueva identidad creada

En la imagen anterior puede apreciarse como se ha generado una nueva identidad una vez se ha completado la autorización.

A continuación, se verá el procedimiento para generar una firma digital con el cliente:

Ilustración 12: Inicio de un proceso de firma digital

La pantalla resultante que puede verse en la imagen anterior es el resultado de seleccionar la opción de firmar de alguna de las identidades. Puede apreciarse que se solicita un documento a firmar, y se enseña el identificador de la firma así como la estimación del tamaó que tendrá la firma una vez esta este realizada (el cálculo se ha realizado en tiempo de registro, ya que el servidor dispone de todos los datos necesarios para estimar este tamaño)

Ilustración 13: Documento a firmar cargado en el cliente

En la imagen anterior, puede apreciarse que una vez seleccionado el documento, aparece la opción de firmar el documento. Aúnque visualmente no puede apreciarse, en el cliente han sucedido unos cuantos eventos:

- Se ha cargado el PDF en la aplicación de navegador.

- Se crean los elementos necesarios definidos por PAdES[2], y se añaden al PDF.

- Se calcula el hash del documeto, para ello se usa el size que se ha calculado y recibido del servidor de firma previamente, y se añade el byte-range al PDF.

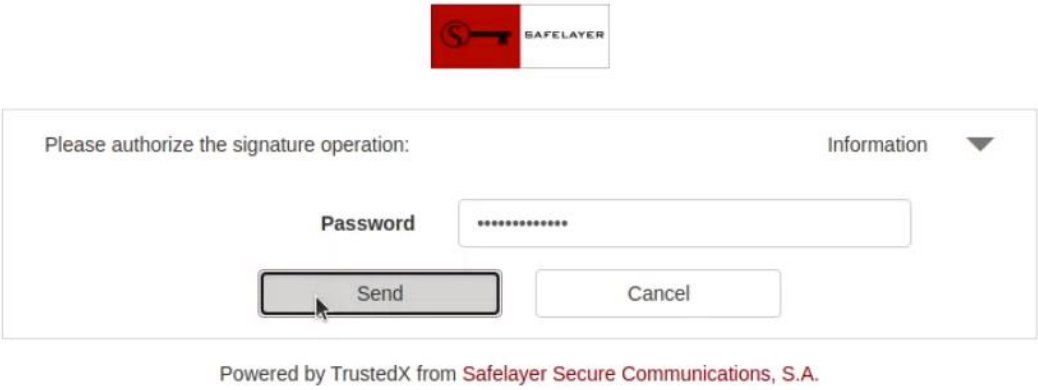

Ilustración 14: Autorización Firma

Lo representado en la imagen anterior sucede cuando se selecciona la opción de firmar PDF, como ya se ha calculado el hash del PDF, este es enviado al servidor de autorización para autorizar la firma de los datos, entonces se completa un flujo de autenticación como el que vimos en el diseño del protocolo.

También puede apreciarse que se solicita, el secreto que ha establecido el usuario para proteger la clave custodiada por el servidor de firma.

Una vez se ha completado el flujo de autenticación, en el servidor de firma recupera por medio del token de autorización la estructura SAD del servidor de autorización y realiza la firma con el componente SAM.

Ilustración 15: Construcción documento en el cliente

Una vez se ha realizado la firma con el servidor, este la envía al cliente, tal y como puede apreciarse en la imagen anterior, donde el cliente la incrusta en el documento PDF, concretamente, en los bytes correspondientes a la firma que ha reservado previamente (campo /Contents rellenado con 0's).

Ilustración 16: PDF firmado

Finalmente una vez se ha construido el documento PDF con la firma, se abre el PDF en otra pestaña, tal y como puede apreciarse en la imagen anterior.

5. Conclusiones

5.1. Objetivos

Con este proyecto se ha conseguido diseñar un protocolo de firma digital en la nube, que cumple con el reglamento eIDAS[1], en el cual el documento que va a firmarse no abandona en ningún momento el dispositivo donde se ha iniciado la firma, ya que el servidor solo recibe el hash del documento que va a firmarse.

En cuanto a los objetivos definidos, han podido cumplirse todos:

- Se ha realizado un estudio de los componentes necesarios, así como de sus requerimientos, que ha servido como fuente de conocimiento de todos los requisitos que deben cumplirse para poder ofrecer un servicio de firma digital cualificada en la nube según lo establecido en eIDAS[1]. Por lo tanto, se ha cumplido este objetivo.

- A partir de los conocimientos obtenidos del estudio del punto anterior, ha sido posible el diseño de un protocolo de firma digital en el cual el documento que va a firmarse permanece en el dispositivo del usuario, con lo cual hemos conseguido preservar la privacidad del documento. Por lo tanto, se ha cumplido este objetivo, que ha resultado en el protocolo de este proyecto.

- Con la implementación del demostrador, se ha demostrado la efectividad del protocolo.

5.2. Planificación y metodología

Se ha intentado respetar lo máximo posible la planificación definida al inicio de este proyecto, sin embargo, la implementación del demostrador ha llevado algo más del definido inicialmente, y su implementación ha tenido que ampliarse y por lo tanto combinarse con el tiempo dedicado a la redacción de la memoria.

La metodología aplicada ha sido la correcta, ya que han podido cumplirse los objetivos, y también porque ha permitido controlar la evolución del proyecto mediante el cumplimiento de las tareas en el tiempo establecido, y también gracias a las reuniones con el director del proyecto donde se repasaba los puntos más importantes a tener en cuenta durante el desarrollo del proyecto.

5.3. Trabajo futuro

El trabajo futuro consiste en la implementación del protocolo en un producto de firma real, aunque en este proyecto se ha implementado un demostrador del protocolo, este no ha sido implementado como un producto final, por lo que aún dispone de un margen de mejora en cuanto a, por ejemplo, optimización y funcionalidad.

6. Glosario

- **eIDAS:** (Electronic IDentification Autentication and trust Services) Se trata de la regulación europea en lo referido a identificación, autenticación y servicios de confianza, entre los cuales se encuentra la firma digital.

- **PAdES:** (PDF Advanced Electronic Signatures) Estándar definido sobre el formato PDF, en el cual se definen los requisitos que debe cumplir una firma PDF para ser una firma PAdES.

- **CAdES:** (CMS Advanced Electronic Signatures) Estándar definido sobre CMS el formato CMS, en el cual se definen los requisitos que debe cumplir una firma CMS para ser una firma CAdES.

- **SAM:** (Signature Activation Module) Componente certificado como Common Criteria EAL4+ con un perfil EN-419241-2, el cual ofrece un servicio de firma digital haciendo uso de un HSM. Ofrece un sistema de protección de claves de los usuarios que garantiza que solo ellos puedan usar sus claves.

- **HSM:** (Hardware Security Module) Componente hardware certificado de alta seguridad que ofrece operaciones criptográficas.

- **QSCD:** (Qualified Electronic Signature Creation Device), Denominación para aquellos componentes que cuentan con una certificación que les permite ofrecer firma digital cualificada.

- **QES:** (Qualified Electronic Signatures) Denominación para aquellas firmas digitales que han sido generadas por un componente QSCD. Es decir, firmas digitales cualificadas.

- **SPA:** (Single Page Application) Aplicación que se ejecuta en un navegador y que no necesita recargar la página con cada interacción que realiza el

usuario, sino que es capaz de gestionar los datos y reescribir secciones de la página dinámicamente.

- **SAD:** (Signature Activation Data) Estructura generada y firmada por un servidor de autorización para autorizar a un usuario la firma sobre unos datos. La firma de esta estructura es verificada por el componente SAM.

- **SKAD:** (Signature Key Activation Data) Estructura generada y firmada por un servidor de autorización para autorizar a un usuario la gestión de una clave de firma custodiada por un servidor. La firma de esta estructura es verificada por el componente SAM.

- **Common Criteria EAL4+:** Certificación que indica que un producto cumple con ciertas medidas de seguridad. Este tipo de certificaciones tienen un reconocimiento internacional.

7. Bibliografía

1. Reglamento eIDAS relativo a la identificación electrónica y los servicios de confianza para las transacciones electrónicas:
    - https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=celex%3A32014R0910

2. Electronic Signatures and Infrastructures (ESI) PDF Advanced Electronic Signature Profiles; PadES Overview – a framerok document por PAdES (Part1):
    - https://www.etsi.org/deliver/etsi_ts/102700_102799/10277801/01.01.01_60/ts_10277801v010101p.pdf

3. Electronic Signatures and Infrastructures (ESI) CMS Advanced Electronic Signatures (CAdES):
    - https://www.etsi.org/deliver/etsi_ts/101700_101799/101733/02.01.01_60/ts_101733v020101p.pdf

4. Electronic Signatures and Infrastructures (ESI) PDF Advanced Electronic Signature Profiles; PAdES Enhanced – PadES-BES and PAdES-EPES Profiles (Part 3):
    - https://www.etsi.org/deliver/etsi_ts/102700_102799/10277803/01.01.02_60/ts_10277803v010102p.pdf

5. Document Management – Portable document format – Part 1 (PDF 1.7):
    - https://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/PDF32000_2008.pdf

6. Crypographic Message Sysntax (CMS):
    - https://tools.ietf.org/html/rfc3852

7. CMS Advanced Electronic Signatures (CAdES):
    - https://tools.ietf.org/html/rfc5126

8. The OAuth 2.0 Authorization Framework
    - https://tools.ietf.org/html/rfc6749

9. OAuth2
    - https://oauth.net/2/

10. Trustworthy Systems Supporting Server Signing Part 2: Protection Profile for QSCD for Server Signing:
    - https://www.commoncriteriaportal.org/files/ppfiles/anssi-cc-pp-2018_02fr_pp.pdf

11. ¿Por qué la certificación Common Criteria es el sello de confianza que debes buscar en tu próxima solución de seguridad?:
    - https://www.opencloudfactory.com/certificacion-common-criteria-solucion-de-seguridad/

12. DNI Electrónico
    - https://firmaelectronica.gob.es/Home/Ciudadanos/DNI-Electronico.html