Sistema de control PKI o Infraestructura de Clave Pública con Python

Breve resumen:

¿Qué es la PKI o Infraestructura de Clave Pública?

La "***Public Key Infrastructure***", por sus siglas en inglés, son un grupo de componentes y servicios informáticos que permiten gestionar, controlar y administrar la tarea de generar, brindar, revocar y validar toda clase de *certificados digitales (* *https://es.wikipedia.org/wiki/Certificado_digital* *).*

En síntesis, es una combinación de hardware y software aplicada en políticas y tareas de seguridad digital. Lo que hace especial a la PKI sobre otros métodos de cifrado, es que puede integrar los certificados digitales junto a la criptografía de la clave pública y las diferentes autoridades de certificación dentro de una misma plataforma.

Su arquitectura está conformada por una infraestructura de confianza que abarca los siguientes actores o componentes:

- *Autoridad de certificación*: (persona juridica con rol de CA)

La PKI permite que las instituciones o autoridades que se encargan de emitir y determinar la validez de los certificados estén incluidos en su estructura.

- *Autoridad de registro*: (persona física con rol de autorizante)

Es en resumen el intermediario entre el usuario final de los certificados y la autoridad de certificación en la tarea de expedir y/o renovar los certificados.

- *Autoridad de validación*: (software dedicado a la gestión de los documentos)

Son aquellos actores que se encargan de centralizar, organizar y controlar la lista de todos los certificados digitales emitidos, vencidos o revocados. Asimismo, permitiendo que toda esta información sea visible para los usuarios.

- *Autoridad de repositorio*: (usualmente un hardware especializado para su resguardo)

El "lugar" donde se almacenan todos los certificados emitidos, los caducados o revocados por cualquier razón.

- *Software y políticas*: (soluciones satélites y sus correspondientes políticas jurídicas según corresponda a las normativas vigentes del estado)

Por último, la PKI integra a todos los productos de software que están destinados para usar los certificados digitales y, por supuesto, aquellas reglas o políticas definidas para la comunicación de la información en este aspecto.

Sin embargo, las PKI tienen un propósito mucho más extenso que el solo cifrar información para certificados digitales.

¿Para qué sirve la PKI?

La confidencialidad de la información es factor fundamental de las transacciones comerciales y la PKI se encarga de garantizar la protección y seguridad de todos los mensajes, su integridad, autenticación y el no repudio. De hecho, la Infraestructura de Clave Pública está presente en varias áreas o herramientas comerciales, por ejemplo:

- *Firma digital*

En el caso de esta tecnología, cuando una persona u organización obtiene un certificado digital para una firma electrónica avanzada, los proveedores certificados (ya sea público o privado en el país de emision) que emiten dicho proceso cuentan con una infraestructura PKI para hacerlo.

Mediante los documentos oficiales y datos biométricos se crea el certificado que, a su vez, es realizado bajo el esquema de PKI para garantizar y proteger la información que se contiene en él.

Ahora bien, la PKI no solo cifra la información para asegurar la identidad del firmante, también:

- *Garantiza el certificado único*: Es decir, evita la creación de más de una llave pública por persona (física o moral). Si se intenta crear otra, teniendo una vigente, este reconocerá que ya existe y no podrá ejecutar el proceso.
- *Valida el certificado*: A través del protocolo OCSP, verifican que el certificado digital no haya sido revocado o esté caducado.

Como ves, el PKI brinda seguridad a las transacciones y ayuda a agilizar los procesos que tienen que ver con las firmas electrónicas. Pero, como ya debes saber, cada país cuenta con diferentes procedimientos para manejar este tipo de componentes.

Referencia: https://www.argentina.gob.ar/firma-digital

Propuesta del proyecto

Generar dos aplicaciones (cliente y servidor) mediante los cuales simularan el flujo de comunicación de una PKI, ejemplo:

- *Lado Cliente*:
    - Aplicación de consola o grafica para firmar digitalmente documentos pdf ya sea utilizando un archivo PKCS12 (*.p12) auto-firmado o bien invocar una web API (ver mas adelante) para firmar el hash del documento. En cualquiera de los dos casos, se previsualizara la firma digital embebida en el documento pdf.
- *Lado Server*:
    - CRUD de clientes y certificados digitales auto-firmados contenidos en PKCS12 mediante una interfaz grafica.
    - CRUD de estados de certificados digitales vigentes.
    - Web API para evaluar estados de OCSP y CRL.

La generación del certificado digital se realiza bajo el estándar x509 (https://es.wikipedia.org/wiki/X.509) en la cual se puede definir los siguientes atributos (entre otros):

- *CRL Distribution Points*: En esta extensión se detalla la ruta del enlace (URL) donde se pueden obtener las Listas de Revocación de Certificados (CRL) correspondientes a la subordinada que emite el certificado final, de este modo se puede automatizar la consulta de las mismas cuando se consulta el certificado.

- *Authority Information Access*: Del mismo modo que en el caso anterior, en esta extensión se mantiene la ruta de acceso (URL) al servicio de validación OCSP de la CA, facilitando también la automatización de la validación pudiendo obtener dicha información del propio certificado.

Ambos atributos serán definidos para consumir nuestras WEB APIs http://localhost:8080/api/crl y http://localhost:8080/api/ocsp respectivamente.

Para mas información, puede consultar: https://www.argentina.gob.ar/estandares-de-firma-digital

Repositorio de datos

La estructura de tablas a considerar serian las siguientes:

- CRUD de usuarios

```
DROP TABLE IF EXISTS `usersign`;
CREATE TABLE IF NOT EXISTS `usersign` (
  `ID` int NOT NULL AUTO_INCREMENT,
  `ALIAS` varchar(255) NOT NULL COMMENT 'ALIAS OR CUIL',
  `NAME` varchar(255) DEFAULT NULL COMMENT 'NAME',
  `LASTNAME` varchar(255) DEFAULT NULL COMMENT 'LAST NAME',
  `FK_CERTSIGN` int NOT NULL COMMENT 'ID CertSign Table',
  `CDATE` datetime DEFAULT CURRENT_TIMESTAMP COMMENT 'Created Date',
  PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4;
```

- CRUD de Certificados digitales

```
DROP TABLE IF EXISTS `certsign`;
CREATE TABLE IF NOT EXISTS `certsign` (
  `ID` int NOT NULL AUTO_INCREMENT,
  `SERIALNUMBER` varchar(255) NOT NULL COMMENT 'SERIAL NUMBER',
  `CERTBASE64` longText DEFAULT NULL COMMENT 'Certificate Base64',
  `PATH` varchar(255) NOT NULL COMMENT 'PATH PKCS12',
  `STATE` int NOT NULL COMMENT 'REVOKE or ACTIVE',
  `CDATE` datetime DEFAULT CURRENT_TIMESTAMP COMMENT 'Created Date',
```

```
  PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4;
```

- LOG de las operaciones realizadas

```
DROP TABLE IF EXISTS `logsign`;
CREATE TABLE IF NOT EXISTS `logsign` (
  `ID` int NOT NULL AUTO_INCREMENT,
  `FK_CERTSIGN` int NOT NULL COMMENT 'ID CertSign Table',
  `HASH` longText DEFAULT NULL COMMENT 'HASH PDF',
  `RESPONSE_STATE` int NOT NULL COMMENT 'OCSP STATE',
  `CDATE` datetime DEFAULT CURRENT_TIMESTAMP COMMENT 'Created Date',
  PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4;
```

Tematicas aplicadas en el proyecto:

- Estructura de carpetas respetando el Patron MVC.
- Programacion orientada a objetos.
- Escritura y lectura de archivos. En este caso, *.p12.
- Importacion de librerias propias y analizando alguna librería externa.
- Escritura de datos en tabla de base de datos.
- Tkinter para las aplicaciones CRUD.
- Servicio web para las APIs.