Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Фізико-технічний інститут

ЛАБОРАТОРНА РОБОТА №1

з дисципліни

«Методи реалізації криптографічних механізмів»

на тему: «Реалізація алгоритмів генерації ключів гібридних криптосистем»

підгруппа 3А

Виконали:

студенти 6 курсу ФТІ

групи ФБ-11мн

Стурчак Максим, Харченко Владислав

Перевірив:

Селюх.П.В.

Реалізація Web-сервісу електронного цифрового підпису.

Існуючі українські аналоги:

- https://ca.diia.gov.ua/sign
- https://privatbank.ua/ru/business/paperless

Основний функціонал веб-сервісу:

1. Створення особистого та відкритого(сертифікату) ключів для електронного цифрового підпису.
2. Збереження сертифікату на сервері у відкритому доступі.
3. Збереження особистого ключа у зашифрованому вигляді в розширенні браузера lastpass.
4. Підпис цільового файлу за допомогою особистого ключа.
5. Пошук сертифікату за ІНН користувача.
6. Перевірка підписаного файлу на відповідність та не втручання 3ї сторони.
7. Видалення сертифікату із веб-застосунку.

Технічні вимоги:

Language	Java
Platform	Java Cryptography Architecture
Signature algorithm	SHA256withDSA
Hash functions	SHA-256
Cipher	RSA
Encoded DB	AES
Secure communication	HTTPS
Creating the Key Pair	key-gen/ KeyPairGenerator
Keystore type	pkcs12
Keystore Cipher type	RSA PKCS12
Encoded key	X.509 or PKCS8

Вимоги до функціоналу

1. **Сертифікати**

- Створення сертифікату за допомогою персональних даних користувача(Ім’я, Прізвище, ІНН, паспортні дані, номер телефону, пошта, пароль захисту ключа). Генерація приватного ключа і збереження його розширенні lastpass в зашифрованому вигляді, доступ до якого можливий лише ввівши пароль захисту

ключа. Збереження особистих даних і сертифікату на сервері веб-застосунку в зашифрованому вигляді. Передача даних від користувача до БД відбувається за допомогою https.
(POST request – передавання всіх введених даних користувачем у форматі JSON)

- Пошук сертифікату за допомогою ІНН користувача
(GET request – фільтр по запиту)
- Відкликання сертифікату за допомогою особистого ключа і паролю захисту ключа.
(DELETE request в разі успішної перевірки приналежності до сертифікату)

2. Підписання файлів

- Підписання файлу за допомогою завантаження цільового файлу та закритого ключа (POST request в разі успішної верифікації ключа з подальшим завантаженням цільового файлу для підписання)

3. Перевірка підписаного файлу

- Перевірка підписаного файлу за допомогою завантаження цільового файлу, підписаного файлу та ІНН користувача, який підписав файл.
(GET request файлу та його перевірка за публічним ключем)

Вимоги до зберігання та передачі даних

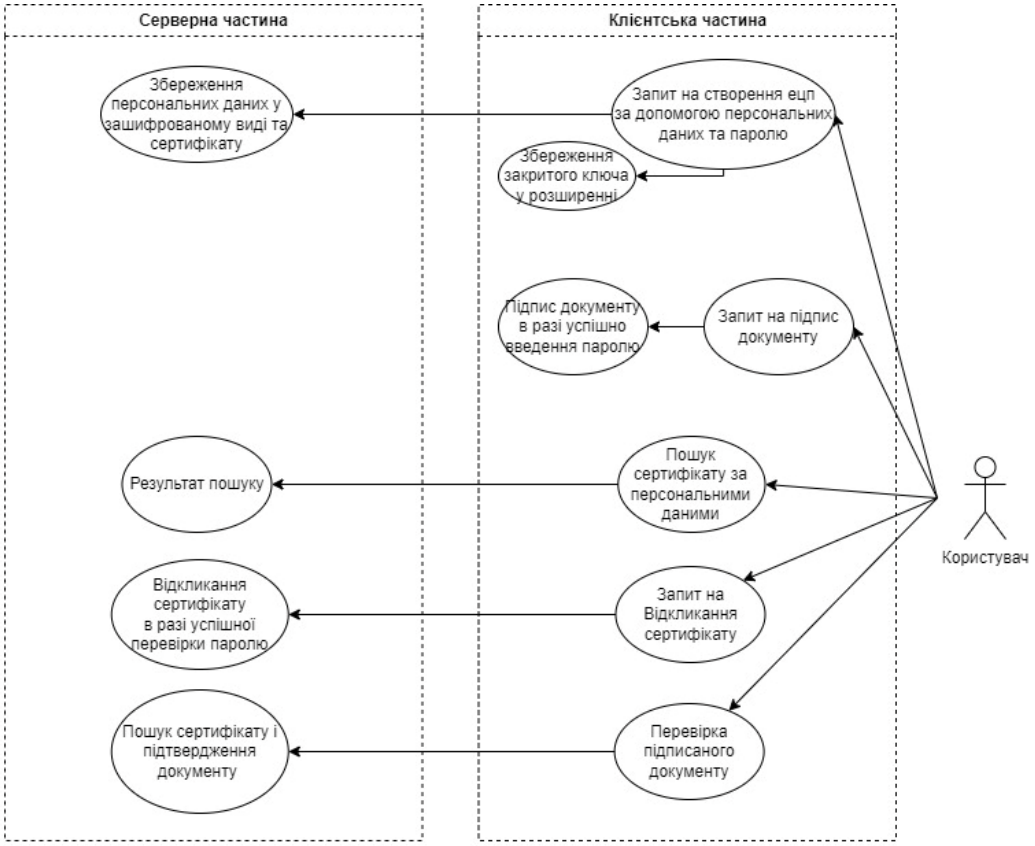
- Зберігання закритих ключів локально у користувача в зашифрованому вигляді у розширенні lastpass. Розширення для сховища ключів захищено паролем, який користувач вводив при створенні ецп.

- Зберігання відкритих ключів(сертифікатів) в Truststore file. cacerts.jks, містить довірені сертифікати сервера додатків, включаючи відкриті ключі для інших об’єктів. Для довіреного сертифіката сервер підтвердив, що відкритий ключ у сертифікаті належить власнику сертифіката.

- Персональні дані користувача зберігаються у зашифрованому вигляді у БД сервера веб-застосунку. Шифрування здійснюються за допомогою AES.

- Передача даних від клієнта до сервера забезпечується за допомогою протоколу https. Для цього необхідно перевести сайт на https за допомогою встановлення SSL-сертифікату на хостинг.

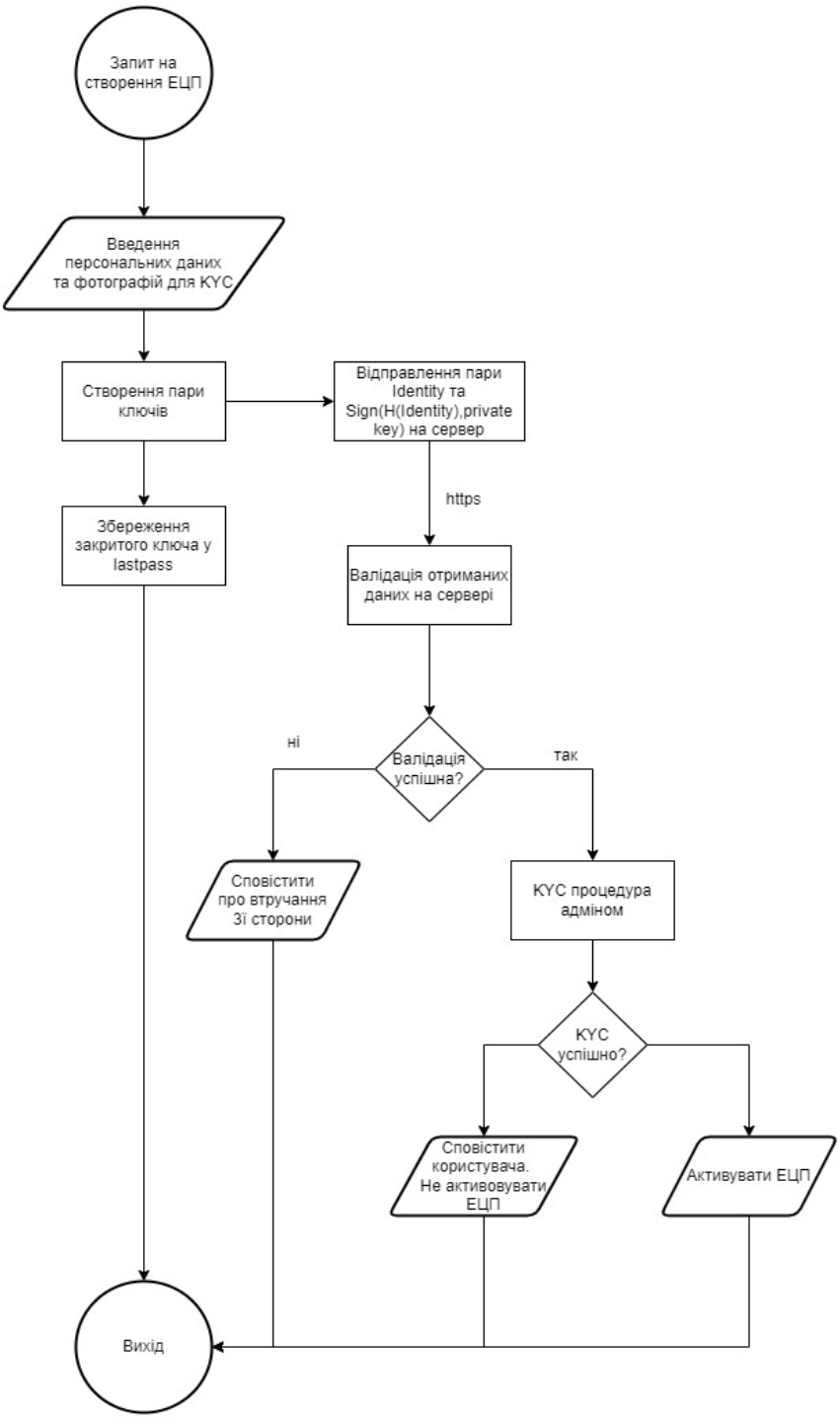
UML діаграма функціоналу



Детальний опис функціоналу

1. **Створення ЕЦП**. Користувач вводить свої персональні дані у відповідні поля та завантажує фотографії паспорта, фото з паспортом для KYC процедури та отримує закритий ключ, який до проходження KYC процедури буде не дійсний. KYC процедуру виконує адмін застосунку. Закритий ключ зберігається у зашифрованому вигляді локально у користувача у розширенні браузеру під паролем який ввів користувач при створенні ЕЦП. Доступ до розширення надається лише після введення пароля.
   На сервер відправляється наступна інформація:
   - Персональні дані + сертифікат = Identity.
   - Захешований Identity підписаний приватним ключем користувача.

   Дана пара потрапляє на сервер по протоколу https для подальшого збереження персональних даних та сертифікату на сервері веб-застосунку у зашифрованому вигляді. Сервер отримує пару даних і сертифікат із Identity для перевірки дійсності отриманих даних. Сервер хешує отриманий Identity та проводить валідацію отриманих даних (Identity, Signature, public key). В разі успішної валідації на сервер зберігається інформація користувача та його сертифікат.



\

Рисунок 1. Схема створення ЕЦП

2. **Пошук сертифікату.** Користувач вводить ІНН особи сертифікат якої хоче знайти. Відбувається запит на сервер по https протоколу. Відбувається розшифрування даних і пошук збігу по ІНН. В разі успішного збігу сервер надсилає клієнту сертифікат, який можна завантажити.

Рисунок 2. Схема пошуку сертифікату

3. **Відкликання сертифікату.** Користувач прикріплює особистий ключ та вводить пароль до цього ключа. У разі правильного введення паролю користувач отримує змогу видалити сертифікат із серверу.

Рисунок 3. Схема роботи відкликання сертифікату

4. **Підписання файлу.** Користувач завантажує цільовий файл та закритий ключ попередньо ввівши пароль захисту для ключа. В разі збігу паролю відбувається підписання документу. В результаті успішного підписання файлу користувач отримує змогу завантажити підписаний файл і надіслати його в будь-якому мессенджері або поштою або ще чимось щоб інший користувач зміг перевірити підписаний файл.

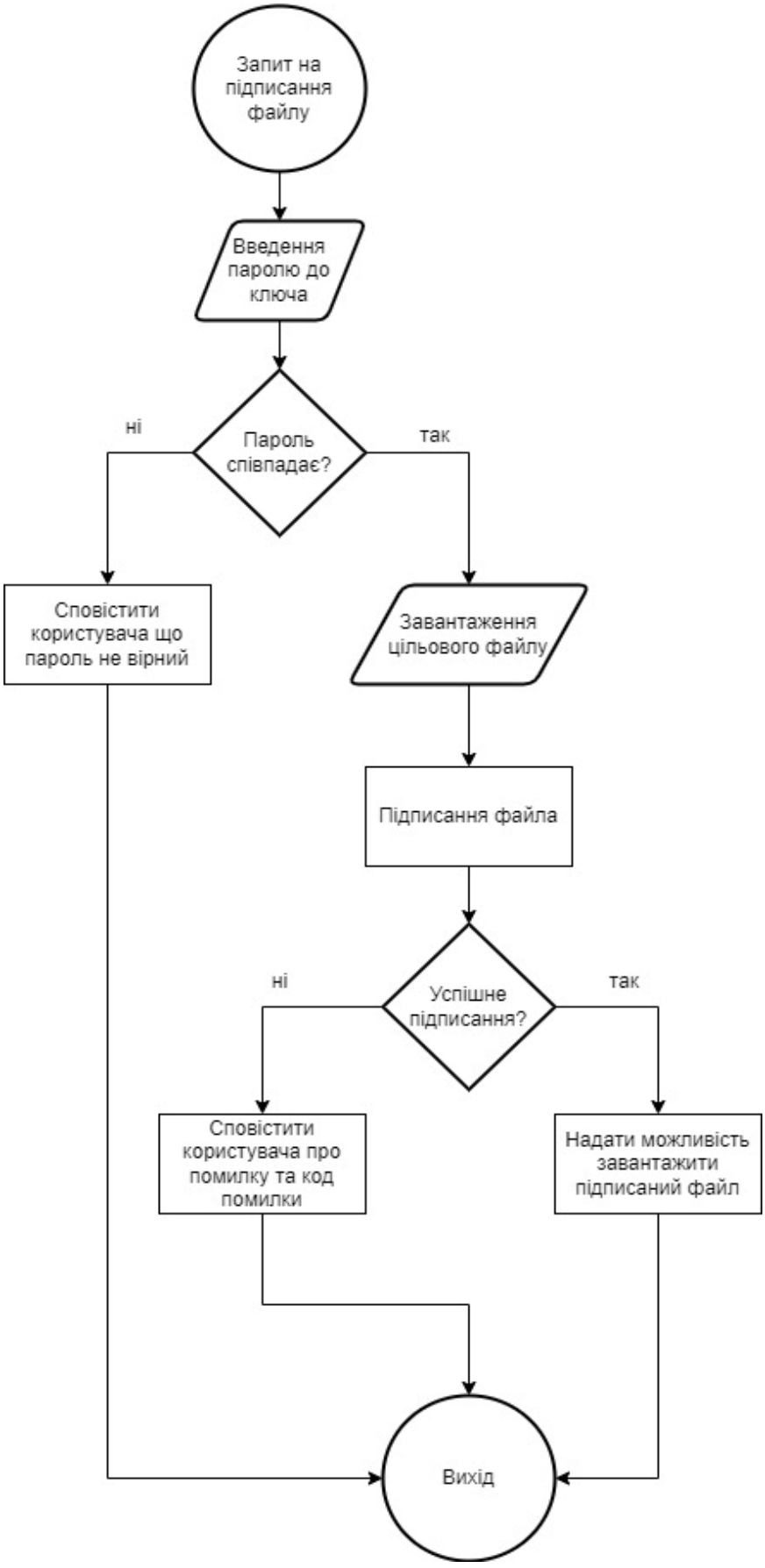


Рисунок 4. Схема підписання сертифікату

5. **Перевірка підписаного файлу.** Користувач А отримав від користувача Б підписаний файл та цільовий файл і хоче перевірити чи дійсно цей файл підписав користувач Б і ніяких змін не було внесено. Він завантажує цільовий файл, підписаний файл та вводить у відповідне поле ІНН користувача Б. В разі успішного пошуку сертифікату по ІНН відбувається перевірка підпису файлу. Після перевірки користувач А отримує сповіщення про успішну або неуспішну перевірку.

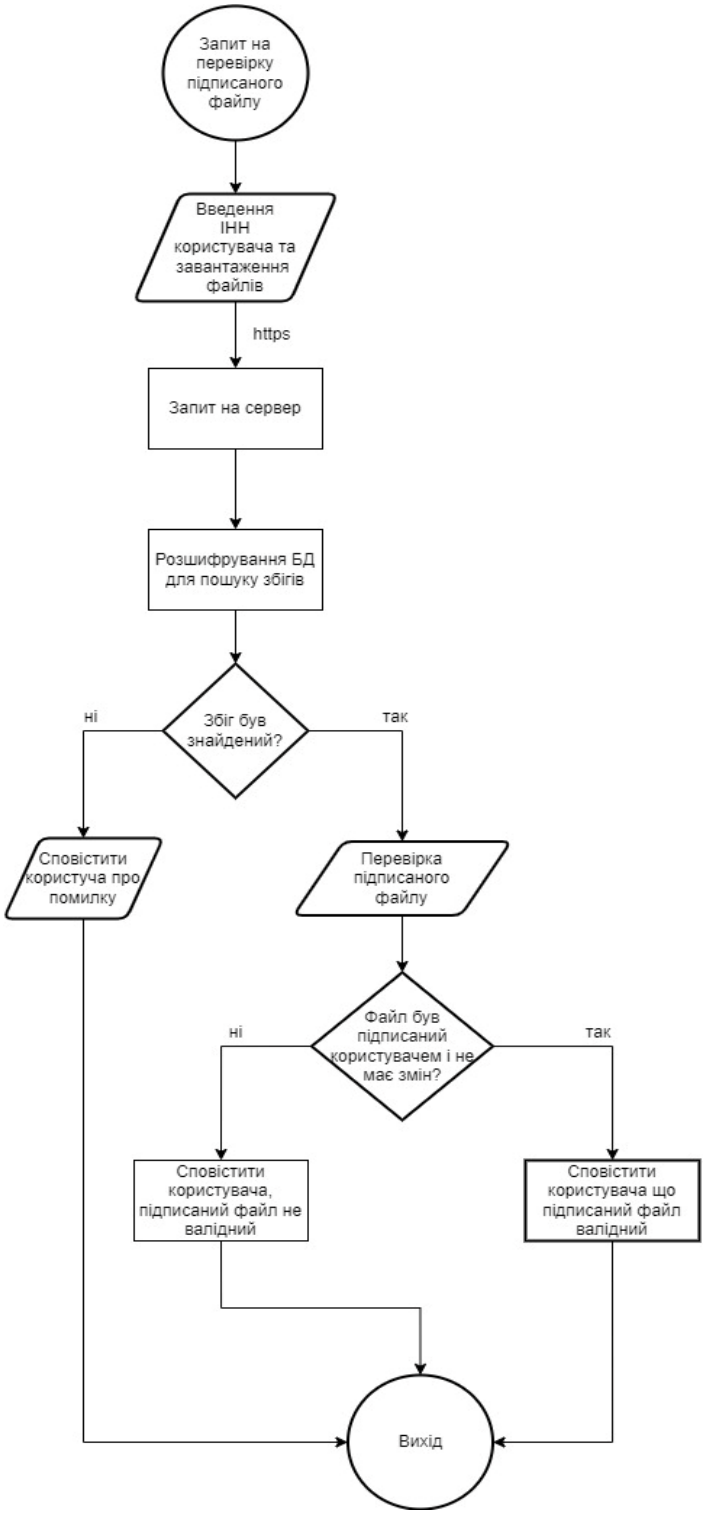


Рисунок 5. Схема перевірки підписаного файлу