Міністерство освіти і науки України
Чернівецький національний університет імені Юрія Федьковича

Інститут фізико-технічних та комп'ютерних наук
Кафедра програмного забезпечення комп'ютерних систем

ЗВІТ

про виконання лабораторної роботи №6
з курсу "Безпека програм та даних"

Тема: Аутентифікація користувачів на основі токенів безпеки

Виконали: Неголюк О.О., Ратушняк М.А.
Перевірив: Остапов С.Е.

Чернівці – 2025

ЗМІСТ

СКЛАД ІНФОРМАЦІЇ У ФАЙЛІ НА ТОКЕНІ БЕЗПЕКИ

Файл токена безпеки зберігає зашифровані дані користувача у такому форматі:

- *salt* – 16 байт випадкових даних, що використовуються для генерації ключа;
- *iv* – 12 байт вектора ініціалізації для режиму AES-GCM;
- *ciphertext* – зашифровані дані у вигляді рядка: `user_id,private_key`;
- *tag* – 16 байт тегу автентичності, що забезпечує цілісність даних.

Таким чином структура файлу є:

$$\text{salt} + \text{iv} + \text{ciphertext} + \text{tag}$$

ОБГОВОРЕННЯ СТІЙКОСТІ ЗАСТОСОВАНИХ ЗАСОБІВ ЗАХИСТУ

Для захисту даних використовується алгоритм *AES-256 у режимі GCM*. Стійкість рішення забезпечується такими особливостями:

- ключ генерується на основі паролю користувача з використанням *PBKDF2-HMAC-SHA256*, 65536 ітерацій;
- *salt* запобігає атакам зі словниками та попередньо обчисленими таблицями (rainbow tables);
- режим *GCM* гарантує не лише конфіденційність, а й цілісність даних завдяки тегу автентичності;
- використання випадкового IV виключає повторне використання одного й того ж ключа для різних повідомлень.

Загалом стійкість системи залежить від складності пароля користувача. При використанні слабких паролів можливі атаки перебором.

ПРОТОКОЛ РОБОТИ АУТЕНТИФІКАЦІЇ

3.1. Вдала аутентифікація

1. Користувач вводить пароль і шлях до токена.
2. Система розшифровує приватний ключ користувача.
3. Сервер надсилає випадковий виклик (challenge).
4. Клієнт підписує виклик приватним ключем.
5. Сервер перевіряє підпис за допомогою публічного ключа користувача.
6. Якщо перевірка пройшла успішно – користувач вважається автентифікованим.

3.2. Невдала аутентифікація

1. Якщо пароль введений неправильно – розшифрування файлу токена завершується з помилкою.
2. Якщо підпис виклику не відповідає публічному ключу користувача – сервер відхиляє аутентифікацію.
3. У обох випадках користувач не отримує доступу до системи.

БЛОК-СХЕМА ТА UML-ДІАГРАМИ

У даному розділі подано графічне представлення роботи системи аутентифікації. Для наочності використано *послідовну діаграму (sequence diagram)*, яка демонструє етапи взаємодії користувача з сервером при автентифікації, а також *діаграму прецедентів* для відображення ролей і сценаріїв використання.

Рисунок 4.1 – Діаграма прецедентів системи аутентифікації

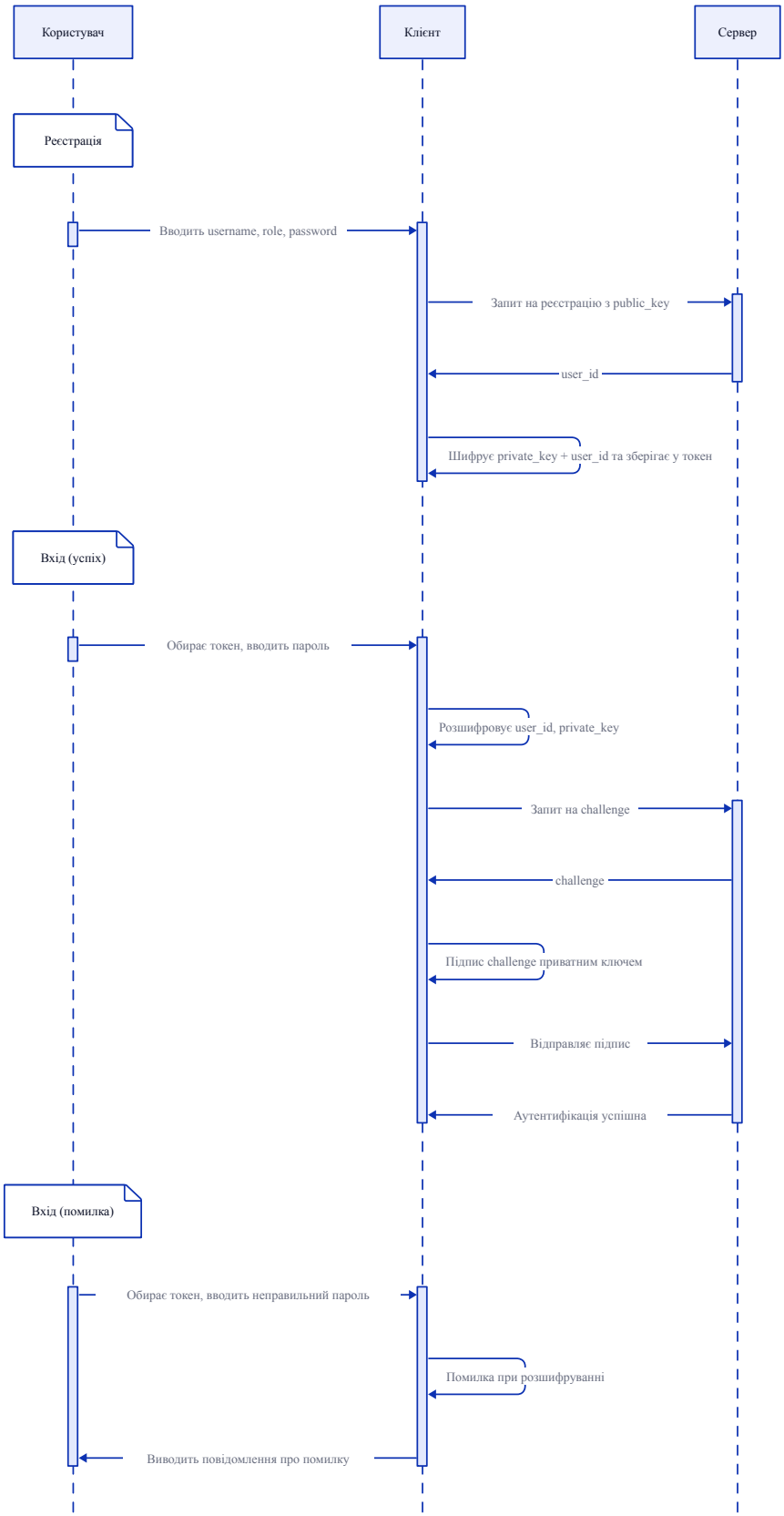

Рисунок 4.2 – Sequence diagram процесу аутентифікації

ВІДПОВІДІ НА КОНТРОЛЬНІ ЗАПИТАННЯ

1. *Що таке токен безпеки?*

Це файл або пристрій, який зберігає секретні ключі користувача у зашифрованому вигляді для автентифікації.

2. *Які алгоритми використовуються для шифрування даних?*

AES-256 у режимі GCM з ключем, отриманим через PBKDF2-HMAC-SHA256.

3. *Які можливі вразливості у системі?*

Використання слабкого пароля користувачем, а також потенційні витоки при некоректному зберіганні токена.

4. *Чим забезпечується цілісність файлу?*

Тегом автентичності GCM, який перевіряється під час розшифрування.

ВИСНОВКИ

У ході лабораторної роботи було реалізовано клієнтську частину системи автентифікації з використанням токенів безпеки. Захист даних забезпечується сучасними криптографічними алгоритмами: PBKDF2-HMAC-SHA256 та AES-256-GCM. Система стійка до більшості поширених атак за умови використання складних паролів. Розроблена архітектура може бути використана як основа для створення безпечних клієнт-серверних застосунків.