Міністерство освіти та науки України

Харківський національний університет радіоелектроніки

Кафедра програмної інженерії

Лабораторна робота №4

з дисципліни: «Безпека програм та даних»

на тему: «Алгоритм обміну ключами Діффі-Хелмана»

Виконав

ст. гр. ПЗПІ-20-1

Бабанін А.К.

Перевірив

доцент кафедри ПІ

Турута О.О.

2023

Мета роботи: Отримати навички безпечного обміну ключами по каналу зв'язку, які надалі можуть бути використані в якомусь алгоритмі шифрування. Реалізувати програмно (на будь-якій мові програмування) роботу алгоритму Діффі-Хеллмана.

Хід роботи

1. Створити програмну реалызацію алгоритму обміну ключами Діффі-Хеллмана.
2. Реалізувати чат на трьох і біольше користувачів.
3. Створити приємний та зрозумілий інтерфейс для перевірки зробленої роботи.

Процес обміну ключами між трьома, або більше учасниками чату виглядає наступним чином: нехай є m користувачів

0. Клієнти під’єднуються до чат-хабу
1. Хаб надсилає кожному кліенту пару g = відкрите просте число та p = відкрите просте число.
2. Кожен кліент обчислює A = відкритий ключ клієнта, та повертає його до хабу
3. Для користувача U_n передається множина відкритий ключів $K = \{k_1, k_2, \dots, k_m\} - \{k_n\}$ для обчислення спільного секрету.

Для взаємодії між клієнтами використовується технологія веб-сокетів та бібліотека. На серверній стороні додатку виконуються наступні операції:

Зберігається інформація про підключених клієнтів;

1. Ведеться облік кількості підключених клієнтів;

2. При підключенні нового клієнта відправляються ініціалізаційні дані (публічні ключі) клієнту

3. Видаляється відключений клієнт зі списку та оновлюється інформація про підключених клієнтів;

4. Дозволяє клієнтам надсилати та отримувати повідомлення.

Лістінг коду наведен далі:

```
function handleKeys(connection, connectionId, data){
    userPublicKeys[connectionId] = data.a;
    if (Object.keys(userPublicKeys).length < 2){

        console.log("Only one user was connected. Waiting...")
    }else{
        console.log("Minimum 2 users were connected. Proceed with keys exchange.")
        for(let key in connections){

            let publicKeys = []

            let toCollect =  Object
                .keys(connections)
                .filter(x => x !== key)
                .forEach(x => publicKeys.push(userPublicKeys[x]));

            let data = {
                users: publicKeys,
            }

            console.log("Sending data to client" + data + " " + key)
            connections[key].send(JSON.stringify(data))
        }

    }
}
```

Чат є анонімний, тому кожен користувач має умовний UUID ідентифікатор у чаті. Для шифрування повідомлень використовується алгоритм AES. Спочатку в анонімному чаті присутні 2 користувачі. Результати роботи застосунку наведені далі:

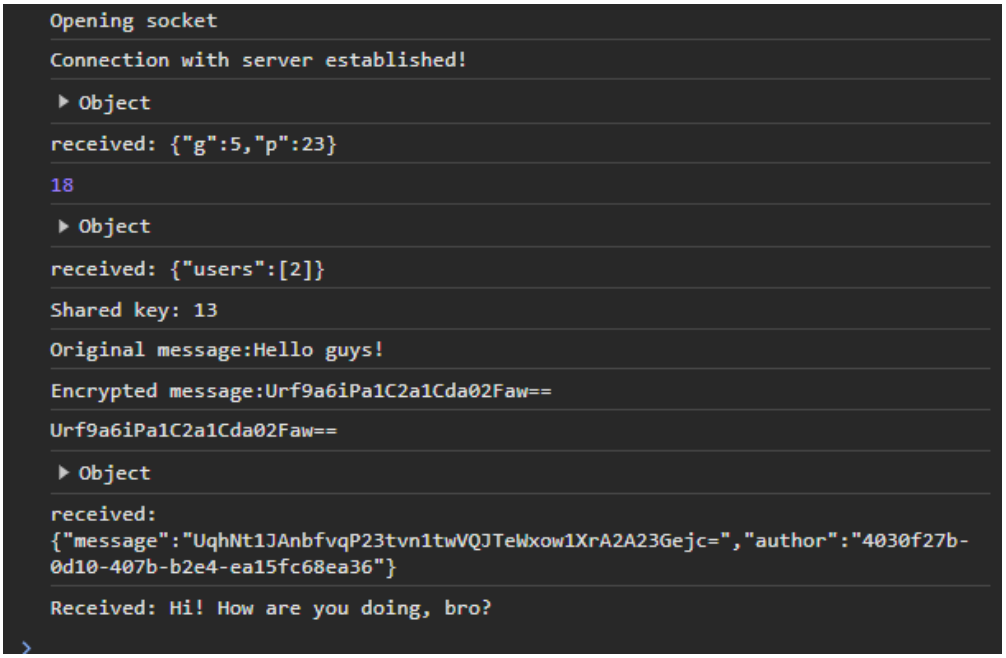

Рисунок 1 – Лог повідомлення першого користувача

Рисунок 2 – Лог повідомлення другого користувача

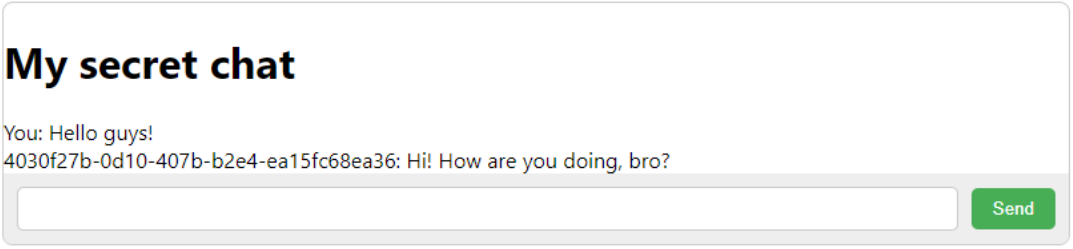

Рисунок 3 – Вигляд чату для першого користувача

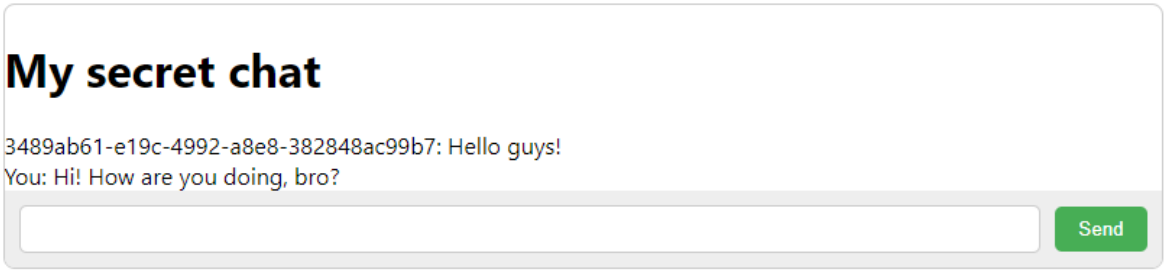

Рисунок 4 – Вигляд чату для другого користувача

Коли під'єднується новий користувач, то відбувається перерахунок ключів:

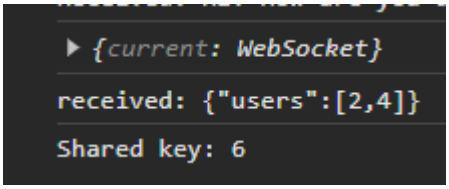

Рисунок 5 – Нові публічні ключі, які отримав перший користувач

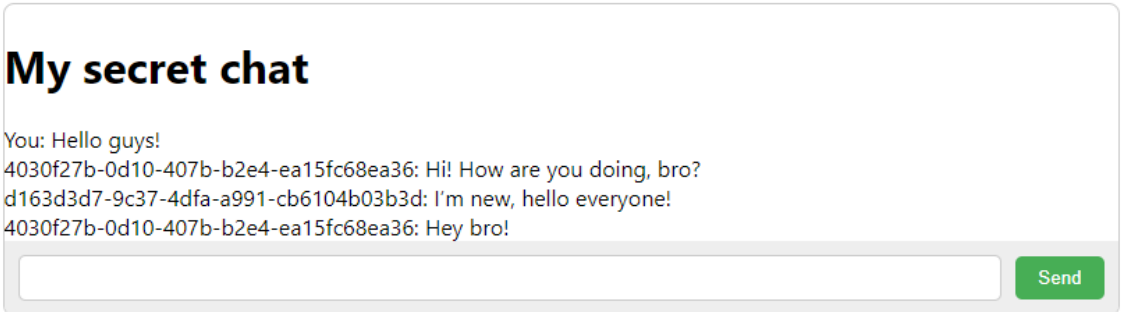

Рисунок 6 – Вигляд чату для першого користувача

Висновки: На даній лабораторній роботі отримав навички безпечного обміну ключами по каналу зв'язку, які надалі можуть бути використані в якомусь алгоритмі шифрування. Реалізував програмно (на мові програмування JavaScript) роботу алгоритму Діффі-Хеллмана