Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА №1

З ДИСЦИПЛІНИ

“ Протоколи і алгоритми електронного голосування”

Виконав:
студент групи ІВ-92
Карпека Д. Ю.

Перевірив:
Нестерчук А. О.

м. Київ – 2022 р.

Тема:
Простий протокол Е-голосування

Мета роботи:
Дослідження простого протоколу Е-голосування

Хід роботи:

Першим було складено клас **Voter**, який описує атрибути і методи, які використовуються voter:

```
class Voter {
    #theirKeys;
    #theirBuileten;
    #hashedBuileten;
        constructor(name, age, theirCVK, message) {
        this.name = name;
        this.age = age;
        this.theirCVK = theirCVK;
        this.message = message;
        this.#theirKeys = { };
        this.#theirBuileten = {
            // signa: { signature and public key of RSA sign } OBJECT
            // buileten: elGamalized with given public key from CVK } STRING
        };
        this.#hashedBuileten = "";
    }

    givePublicKey() {
        return this.#theirKeys.publicRSAKey;
    }

    #generatePublicKey() {
        const { publicKey, privateKey } = crypto.generateKeyPairSync("rsa", {
modulusLength: 2048});
        this.#theirKeys.publicRSAKey = publicKey;
        this.#theirKeys.privateRSAKey = privateKey;
    }

    #signBuileten() {
        this.#generatePublicKey();
        const verifiableData = this.name;
        const signature = crypto.sign("sha256", Buffer.from(verifiableData), {
            key: this.#theirKeys.privateRSAKey,
            padding: crypto.constants.RSA_PKCS1_PSS_PADDING,
```

```
        });

          this.#theirBuileten.signa = {
            theirSign: signature,
            theirPKey: this.#theirKeys.publicRSAKey
          }
    }

    #elGamalizeBuileten(channels) {
        const hashedMessage = channels.ch1.encrypt(this.message,
channels.ch2.pubKey);
        this.#hashedBuileten = hashedMessage;
        this.#theirBuileten.buileten = this.#hashedBuileten;
    }

    sendBuileten(channels) {
        this.#signBuileten();
        this.#elGamalizeBuileten(channels);
        return this.#theirBuileten;
    }
}
```

З метою захисту, методи та змінні відділились на 2 групи: захищені та незахищені. Це пов’язано, перш за все, із реалізацією протоколу безпечного голосування, який би задовільняв умови (1) та (2) ідеального протоколу голосування.

Наступним кроком була розробка класу CVK. Це наша виборча комісія

```
class CVK {
    #listOfVoters;
    constructor(nameOfCVK) {
        this.nameOfCVK = nameOfCVK;
        this.listOfCandidates = [
            {
                name: "Ilon Musk, who support Ukraine",
                votes: 0
            },
            {
                name: "Ilon Musk, who support russia",
                votes: 0
            }
        ];
        this.#listOfVoters = [
            {
                name: "Maria Andrushko",
```

```
            age: 18,
            channels: {
                ch1: ElGamal(Alphabet, 19),
                ch2: ElGamal(Alphabet, 19)
            }
        },
        {
            name: "Pavlo Stonkevych",
            age: 21,
            channels: {
                ch1: ElGamal(Alphabet, 27),
                ch2: ElGamal(Alphabet, 27)
            }
        },
        {
            name: "Yakym Galayda",
            age: 19,
            channels: {
                ch1: ElGamal(Alphabet, 26),
                ch2: ElGamal(Alphabet, 26)
            }
        },
        {
            name: "Oksana Styslo",
            age: 42,
            channels: {
                ch1: ElGamal(Alphabet, 15),
                ch2: ElGamal(Alphabet, 15)
            }
        },
        {
            name: "Vasylyna Prybylo",
            age: 67,
            channels: {
                ch1: ElGamal(Alphabet, 18),
                ch2: ElGamal(Alphabet, 18)
            }
        },
        {
            name: "Vadym Detec",
            age: 31,
            channels: {
                ch1: ElGamal(Alphabet, 24),
                ch2: ElGamal(Alphabet, 24)
            }
        },
        {
```

```
                name: "Anna Yaroslavna",
                age: 990,
                channels: {
                    ch1: ElGamal(Alphabet, 11),
                    ch2: ElGamal(Alphabet, 11)
                }
            },
            {
                name: "Dmytrii Karpeka",
                age: 20,
                channels: {
                    ch1: ElGamal(Alphabet, 62),
                    ch2: ElGamal(Alphabet, 62)
                }
            }];
    }


    givePublicKey(voterName) {
        let indexOfName;
        const canFindName = this.#listOfVoters.find((v, i) => {
            indexOfName = i;
            return v.name === voterName;
        });

        const checkedStatus = this.#listOfVoters[indexOfName].status === "checked";

        if (canFindName) {
            return this.#listOfVoters[indexOfName].channels;
        } else if (!canFindName) {
            console.warn(`Sorry, but you cannot vote at this CVK, ${voterName}`);
            return 0;
        } else if (checkedStatus) {
            console.warn(`Don't try to fool us, you're trying to vote second time,
${voterName}`);
            return 1;
        }
    };

    getBuileten(buileten) {
        const encryptedMessage = buileten.buileten;

        const profileOfVoter = this.#listOfVoters.find((voterN, i) => {
            const isVerified = crypto.verify(
                "sha256",
                Buffer.from(voterN.name),
                {
```

```
                    key: buileten.signa.theirPKey,
                    padding: crypto.constants.RSA_PKCS1_PSS_PADDING,
                },
                buileten.signa.theirSign
            )

            if (isVerified) {
                this.#listOfVoters[i].status = "checked";
            }

            return isVerified;
        });

        const decryptedMessage =
profileOfVoter.channels.ch2.decrypt(buileten.buileten);


        const favourite = this.listOfCandidates.find((candidate) => {
            if (candidate.name === decryptedMessage) {
                profileOfVoter.voteFor = decryptedMessage;
                candidate.votes++;
            }
            return candidate.name === decryptedMessage;
        });

        if (!favourite) {
            console.warn(`We don't have this option, please pay attention,
${profileOfVoter.name}!`);
            return 3;
        }
    }

    finalResults() {


        console.log(`
        ~~~~~
        ${this.listOfCandidates[0].name} has scored
${this.listOfCandidates[0].votes}!
        ${this.listOfCandidates[1].name} has scored
${this.listOfCandidates[1].votes}!

        The winner is ${this.listOfCandidates[0].votes >=
this.listOfCandidates[1].votes ? this.listOfCandidates[0].name :
this.listOfCandidates[1].name}!

        ~~~~~
```

```
        `)
        return this.#listOfVoters.length - this.#listOfVoters.filter((persona) => {
            return persona.status === "checked"
        }).length
    }
}
```

Тут присутній захищений список виборців та методи видачі публічного ключа для передачі повідомлення виборця, валідації підписів (та аутентифікації виборців за цими підписами), обробка голосів та публікація фінальний результатів.

Глобальна змінна statistics у якій ведеться облік про порушення на виборчій дільниці:

```
let statistic = {
    "Явка на вибори:" : 0,
    "Не проголосувало:": 0,
    "Проголосувало неправильно": 0,
    "Виборець не має права голосувати": 0,
    "Виборець хоче проголосувати повторно": 0,
};
```

Функція для фіксації цих порушень:

```
function eVote(voter, cvk) {
    const pk = cvk.givePublicKey(voter.name);
    if (pk === 0) {
        statistic["Виборець не має права голосувати"]++;
        statistic["Явка на вибори:"]++;
        return 0;
    } else if (pk === 1) {
        statistic["Виборець хоче проголосувати повторно"]++;
        statistic["Явка на вибори:"]++;
        return 0;
    }

    const sendedBuileten = voter.sendBuileten(pk);

    const gottenBuileten = cvk.getBuileten(sendedBuileten);

    if (gottenBuileten === 3) {
        statistic["Проголосувало неправильно"]++;
        statistic["Явка на вибори:"]++;
        return 0;
    }
```

```
    statistic["Явка на вибори:"]++;
}
```

Driven code та entry point програми у якій присутні екземпляри вище перелічених класів:

```
function eVoting() {
    let CVK1 = new CVK("CVK#1");

    let voters = [
        new Voter("Maria Andrushko", 18, "CVK#1", "Ilon Musk, who support
Ukraine"),
        new Voter("Dodik from Kremlin", 4, "CVK#1", "Ilon Musk, who support
russia"),
        new Voter("Pavlo Stonkevych", 21, "CVK#1", "Ilon Musk, who support
Ukraine"),
        new Voter("Yakym Galayda", 19, "CVK#1", "Ilon Musk, who support Ukraine"),
        new Voter("Oksana Styslo", 42, "CVK#1", "Ilon Musk, who support Ukraine"),
        new Voter("Vadym Detec", 31, "CVK#1", "Ilon Musk is very toxic, I don't
like him at all"),
        new Voter("Dmytrii Karpeka", 20, "CVK#1", "I agree with Vadym on his
statement"),
        new Voter("Dmytrii Karpeka", 20, "CVK#1", "I have another option now...")
    ]



    voters.map((voter) => eVote(voter, CVK1));
    statistic["Не проголосувало:"] = CVK1.finalResults();

    Object.entries(statistic).map(statement => console.log(statement[0] + " " +
statement[1]))
}
```

Також у цій же функції відбувається публікація статистики про порушення.

Результат:

```
PS D:\Studying\7\ПАЕГ\lw1> node .\start.js
Don't try to fool us, you're trying to vote second time, Maria Andrushko
Sorry, but you cannot vote at this CVK, Dodik from Kremlin
We don't have this option, please pay attention, Vadym Detec!
We don't have this option, please pay attention, Dmytrii Karpeka!
Don't try to fool us, you're trying to vote second time, Dmytrii Karpeka

        ~~~~~~
        Elon Musk, who support Ukraine has scored 4!
        Elon Musk, who support russia has scored 1!

        The winner is Elon Musk, who support Ukraine!

        ~~~~~~

Явка на вибори: 8
Не проголосувало: 1
Проголосувало неправильно 2
Виборець не має права голосувати 1
Виборець хоче проголосувати повторно 2
```

Дослідження протоколу:

1. Перевірити чи можуть голосувати ті, хто не має на це права.

   Ні, на основі складеного алгоритму і його моделюванні виявлено, що виборці не можуть голосувати, якщо не мають на це права.

   (стрічка 2)

2. Перевірити чи може виборець голосувати кілька разів.

   Ні, виборча комісія бачить це правопорушення і вносить відповідні дані у захищений атрибут свого класу. (стрічка 1)

3. Чи може хтось (інший виборець, ЦВК, стороння людина) дізнатися за кого проголосували інші виборці?

   Ні, так, ні. Стороння людина або інший виборець не має доступу до захищених даних ЦВК. ЦВК має доступ до всіх даних, адже із цифровим підписом виборця бачить ім’я виборця і використовує ім’я, щоб

доступитись до сутності виборця, де вказана пара приватного і публічного ключа для дешифрування алгоритмом ElGamal.

```
275
276     // Дослідження
277     try {
278         CVK1.listOfVoter();
279     } catch (e) {
280         console.log("Somebody wanted to stole our database!");
281 }
282 }
283
```

PROBLEMS 1 OUTPUT TERMINAL JUPYTER DEBUG CONSOLE powershell

```
        The winner is Elon Musk, who support Ukraine!

        ~~~~~

Явка на вибори: 8
Не проголосувало: 1
Проголосувало неправильно 2
Виборець не має права голосувати 1
Виборець хоче проголосувати повторно 2
Somebody wanted to stole our database!
PS D:\Studying\7\ПАЕГ\lw1>
```

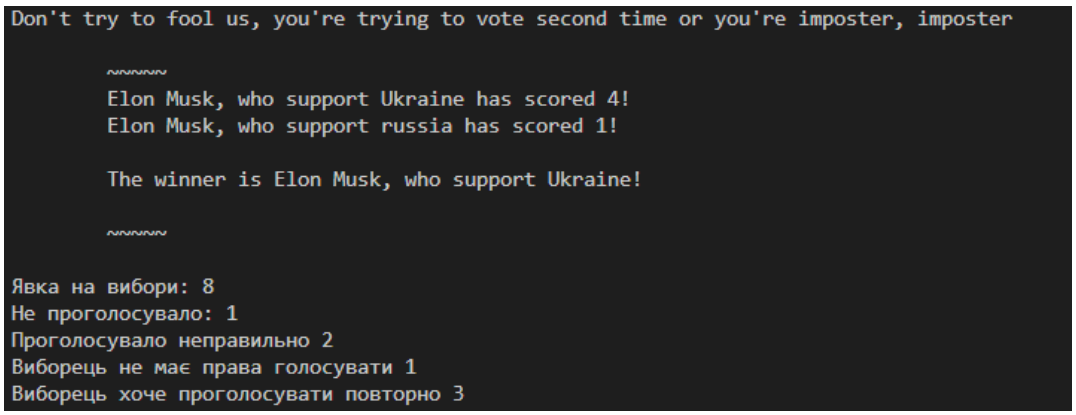
4. Перевірити чи може інший виборець чи стороння людина проголосувати замість іншого зареєстрованого виборця.

   Нормальні умови:

```
        ~~~~~
        Elon Musk, who support Ukraine has scored 4!
        Elon Musk, who support russia has scored 1!

        The winner is Elon Musk, who support Ukraine!

        ~~~~~

Явка на вибори: 8
Не проголосувало: 1
Проголосувало неправильно 2
Виборець не має права голосувати 1
Виборець хоче проголосувати повторно 2
Somebody wanted to stole our database!
PS D:\Studying\7\ПАЕГ\lw1>
```

   Дослідження:

```
class Imposter extends Voter {
        constructor(name, age, cvk, message) {
            super(name, age, cvk, message)
            this.name = "imposter";
        }
    }

new Imposter("Anna Yaroslavna", 990, "CVK#1", "Elon Musk, who support russia")
```



5. Чи може хтось (інший виборець, ЦВК, стороння людина) таємно змінити голос в бюлетені?

   Ні, ні, ні. Алгоритм RSA ЕЦП доступний тільки для виборця, ЦВК таке не вміє.

6. Чи може виборець перевірити, що його голос врахований при підведенні кінцевих результатів?

   Ні, не може. Тільки ЦВК має доступ до захищених даних.

Висновок

У лабораторній роботі було змодельовано простий алгоритм і досліджено його слабкі та сильні сторони. Також було використано криптографічні алгоритми для шифрування даних і безпечної передачі них між різними сутностями. Було зосереджено увагу на використанні ООП, адже ця парадигма дозволяє інкапсульовувати необхідні механіки при реалізації Е-Голосування.