Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

З лабораторної роботи № 3 з дисципліни
«Протоколи й алгоритми електронного голосування»

“**Протокол Е-голосування з двома виборчими комісіями**”

Виконав(ла) *ІП-13 Бабіч Денис* _____

(шифр, прізвище, ім'я, по батькові)

Перевірив *Нестерук А. О.* _____

(посада, прізвище, ім'я, по батькові)

Київ 2024

ЛАБОРАТОРНА РОБОТА № 3

Тема роботи: Протокол Е-голосування з двома виборчими комісіями.

Мета роботи: Дослідити протокол Е-голосування з двома виборчими комісіями.

Основне завдання:

Змоделювати протокол Е-голосування з двома виборчими комісіями будь-якою мовою програмування та провести його дослідження. Для кодування повідомлень використовувати метод Ель-Гамаля, для реалізації ЕЦП використовувати алгоритм DSA.

Умови: В процесі голосування повинні приймати участь не менше 2 кандидатів та не менше 4 виборців. Повинні бути реалізовані сценарії поведінки на випадок порушення протоколу (виборець не проголосував, проголосував неправильно, виборець не має права голосувати, виборець хоче проголосувати повторно, виборець хоче проголосувати замість іншого виборця та інші).

На основі змодельованого протоколу провести його дослідження (Аналіз повинен бути розгорнутим та враховувати всі можливі сценарії подій під час роботи протоколу голосування):

1. Перевірити чи можуть голосувати ті, хто не має на це права.
2. Перевірити чи може виборець голосувати кілька разів.
3. Чи може хтось (інший виборець, ВК, стороння людина) дізнатися за кого проголосували інші виборці?
4. Перевірити чи може інший виборець чи стороння людина проголосувати замість іншого зареєстрованого виборця.
5. Чи може хтось (інший виборець, ВК, стороння людина) таємно змінити голос в бюлетені?
6. Чи може виборець перевірити, що його голос врахований при підведенні кінцевих підсумків?

Виконання завдання:

Реєстраційне бюро представлено відповідним класом (рис. 1.1) та має метод для реєстрації виборця у випадку наявності прав для голосування.

```python
class RegistrationBureauController:

    def __init__(self) -> None:
        self._registered_users = dict()

    def get_voters(self) -> list[int]:
        return list(self._registered_users.values())

    def register(self, user: User) -> VoterController:
        if user is None:
            raise ValueError("user cannot be None")

        print(f"{Fore.YELLOW}REGISTRATION{Style.RESET_ALL} #{user.get_id()} | ", end = '')

        print(f"{Fore.YELLOW}STATUS{Style.RESET_ALL}: ", end = '')

        if not user.get_is_able_to_vote():
            print(f"{Fore.RED}FAILURE{Style.RESET_ALL} (User is not able to vote)")
            return None

        if user in self._registered_users.keys():
            print(f"{Fore.RED}FAILURE{Style.RESET_ALL} (User has been already registered)")
            return None

        registration_id = uuid.uuid4().int
        self._registered_users[user] = registration_id
        voter_controller = VoterController(registration_id)

        print(f"{Fore.GREEN}SUCCESS{Style.RESET_ALL} ({registration_id})")

        return voter_controller
```

Рисунок 1.1 – Клас реєстраційного бюро

```python
class VoterController:

    def __init__(self, registration_id: int) -> None:
        self._custom_id = uuid.uuid4().int
        self._registration_id = registration_id
        self._dsa_private_key = dsa.generate_private_key(key_size = 1024)
        self._dsa_public_key = self._dsa_private_key.public_key()
        self._elgamal_public_key, self._elgamal_private_key = elgamal.Elgamal.newkeys(10)    "newkeys": Unknown word.

    def get_custom_id(self) -> int:
        return self._custom_id

    def get_registration_id(self) -> int:
        return self._registration_id

    def get_dsa_public_key(self) -> dsa.DSAPublicKey:
        return self._dsa_public_key

    def get_elgamal_private_key(self) -> elgamal.PrivateKey:
        return self._elgamal_private_key

    def vote(self, candidate: Candidate) -> VoteDTO:
        vote_payload = VotePayload(candidate.get_id())
        signature = self._dsa_private_key.sign(hash(vote_payload.get_candidate_id()).to_bytes(length = 32), hashes.SHA256())
        vote_dto = VoteDTO(self._custom_id, self._registration_id, vote_payload, signature)
        encrypted_vote_dto = VoteCipherUtility.encrypt(vote_dto, self._elgamal_public_key)
        return encrypted_vote_dto
```

Рисунок 1.2 – Клас виборця

Виборець представленик класом VoterController, у якому є поля dsa-ключів, ключів Ель-Гамаля.

Клас ВК представлений завдяки ElectionCommission, у якому метод register_vote перевіряє розшифровує повідомлення, валідує сам голос і зараховує його у кінцевому результаті.

```python
class ElectionCommissionController:

    def __init__(self, candidates: list[Candidate], voters_registration_ids: list[int]) -> None:
        self._public_results = dict()
        self._voters = copy.deepcopy(voters_registration_ids)
        self._results = {candidate.get_id(): 0 for candidate in candidates}

    def register_vote(self, vote: VoteDTO, dsa_public_key: dsa.DSAPublicKey, elgamal_private_key: elgamal.PrivateKey) -> None:
        if vote is None:
            raise ValueError("vote cannot be None")

        if dsa_public_key is None:
            raise ValueError("dsa_public_key cannot be None")

        try:
            decrypted_vote = VoteCipherUtility.decrypt(vote, elgamal_private_key)
        except Exception:
            print(f"{Fore.RED}FAILURE{Style.RESET_ALL} (Invalid encryption)")
            return

        print(f"{Fore.YELLOW}PROCESSING{Style.RESET_ALL} #{decrypted_vote.get_voter_registration_id()} | ", end = '')

        if not self._verify_signature(vote, dsa_public_key):
            return

        if not self._validate_vote(decrypted_vote):
            return

        print(f"{Fore.GREEN}SUCCESS{Style.RESET_ALL} | ", end = '')

        self._voters.remove(decrypted_vote.get_voter_registration_id())
        self._results[decrypted_vote.get_vote_payload().get_candidate_id()] += 1
        self._public_results[decrypted_vote.get_voter_custom_id()] = decrypted_vote.get_vote_payload().get_candidate_id()

        print(f"{Fore.YELLOW}DONE{Style.RESET_ALL}")
```

Рисунок 1.3 – Метод register_vote класу ElectionCommission

```python
class VotePayload:

    def __init__(self, candidate_id: int) -> None:
        self._timestamp = datetime.now()
        self._candidate_id = candidate_id

    def get_candidate_id(self) -> int:
        return self._candidate_id

    def __hash__(self) -> int:
        return hash(self._candidate_id) ^ hash(self._timestamp)

    def __eq__(self, other: object) -> bool:
        if not isinstance(other, VotePayload):
            return False
        return (self._candidate_id == other._candidate_id and self._timestamp == other._timestamp)

class VoteDTO:

    def __init__(self, voter_custom_id: int, voter_registration_id: int, vote_payload: VotePayload, signature: bytes) -> None:
        self._signature = signature
        self._vote_payload = vote_payload
        self._voter_custom_id = voter_custom_id
        self._voter_registration_id = voter_registration_id

    def get_signature(self) -> bytes:
        return self._signature

    def get_voter_custom_id(self) -> int:
        return self._voter_custom_id

    def get_voter_registration_id(self) -> int:
        return self._voter_registration_id

    def get_vote_payload(self) -> VotePayload:
        return self._vote_payload
```

Рисунок 1.4 – Класи-моделі представлення голосів

```python
candidate_1 = Candidate()
candidate_2 = Candidate()

user_0 = User(False)
user_1 = User(True)
user_2 = User(True)
user_3 = User(True)
user_4 = User(True)

registration_bureau = RegistrationBureauController()

voter_0 = registration_bureau.register(user_0)
voter_1 = registration_bureau.register(user_1)
voter_2 = registration_bureau.register(user_2)
voter_3 = registration_bureau.register(user_3)
voter_4 = registration_bureau.register(user_4)

election_commission = ElectionCommissionController([candidate_1, candidate_2], registration_bureau.get_voters())

# Without registration
voter_0 = VoterController(101010)
vote_0 = voter_0.vote(candidate_1)
election_commission.register_vote(vote_0, voter_0.get_dsa_public_key(), voter_0.get_elgamal_private_key())

# OK
vote_1 = voter_1.vote(candidate_1)
election_commission.register_vote(vote_1, voter_1.get_dsa_public_key(), voter_1.get_elgamal_private_key())

# Second attempt
vote_1 = voter_1.vote(candidate_1)
election_commission.register_vote(vote_1, voter_1.get_dsa_public_key(), voter_1.get_elgamal_private_key())

# OK
vote_2 = voter_2.vote(candidate_1)
election_commission.register_vote(vote_2, voter_2.get_dsa_public_key(), voter_2.get_elgamal_private_key())

# Invalid signature
vote_3 = voter_3.vote(candidate_1)
election_commission.register_vote(vote_3, voter_2.get_dsa_public_key(), voter_3.get_elgamal_private_key())

# Invalid candidate
vote_4 = voter_4.vote(Candidate())
election_commission.register_vote(vote_4, voter_4.get_dsa_public_key(), voter_4.get_elgamal_private_key())
```

Рисунок 1.5 – Сценарії виконання

```
REGISTRATION #0 | STATUS: FAILURE (User is not able to vote)
REGISTRATION #1 | STATUS: SUCCESS (506079487836516825058973414611330147 08)
REGISTRATION #2 | STATUS: SUCCESS (154917629185035302194476475945646976291)
REGISTRATION #3 | STATUS: SUCCESS (290100583513486194977924567229700935227)
REGISTRATION #4 | STATUS: SUCCESS (382451044635610247857072664604779 6239)
PROCESSING #101010 | SIGNATURE_VALIDATION: SUCCESS | VOTE_VALIDATION: FAILURE (Invalid voter)
PROCESSING #506079487836516825058973414611330147 08 | SIGNATURE_VALIDATION: SUCCESS | VOTE_VALIDATION: SUCCESS | SUCCESS | DONE
PROCESSING #506079487836516825058973414611330147 08 | SIGNATURE_VALIDATION: SUCCESS | VOTE_VALIDATION: FAILURE (Invalid voter)
PROCESSING #154917629185035302194476475945646976291 | SIGNATURE_VALIDATION: SUCCESS | VOTE_VALIDATION: SUCCESS | SUCCESS | DONE
PROCESSING #290100583513486194977924567229700935227 | SIGNATURE_VALIDATION: FAILURE (Invalid signature)
PROCESSING #382451044635610247857072664604779 6239 | SIGNATURE_VALIDATION: SUCCESS | VOTE_VALIDATION: FAILURE (Invalid candidate)

RESULTS

231180092717809350695554769401002241237: 1
748214901624954387909730180604927824 83: 1

Candidate #1: 2 vote(s)
Candidate #2: 0 vote(s)

WINNER is candidate #1 with 2 vote(s).
```

Рисунок 1.6 – Результати виконання

Дослідження протоколу:

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

   Ні, не можуть, оскільки відбувається попередня валідація виборця за допомогою реєстраційного бюро. Навіть якщо такий виборець спробує надіслати свій бюлетень, то його голос не буде зараховано, оскільки його немає у списку від бюро. Звісно, що не можна відкидати варіант з шахраюванням з боку реєстраційного бюро, або виборчої комісії.
2. Перевірити чи може виборець голосувати кілька разів.

   Ні, не може, оскільки відбувається попередня перевірка на те, чи присутній реєстраційний номер кандидата у відповідному списку від бюро. Звісно, що не можна відкидати варіант з шахраюванням з боку виборчої комісії.
3. Чи може хтось (інший виборець, ВК, стороння людина) дізнатися за кого проголосували інші виборці?

   Так, оскільки після виборів виводиться список власних айді виборців та вміст їх бюлетенів, але для цього потрібно знати якому виборцю належить який айді. Стосовно самої виборчої комісії, то особистість виборця прихована за спеціальним айді виборця, оскільки остаточний бюлетень присилається без прив’язки до особи, але реєстраційне бюро зберігає повноту інформація про голосувальники та його особистість.
4. Перевірити чи може інший виборець чи стороння людина проголосувати замість іншого зареєстрованого виборця.

   Це можливо зробити, якщо вгадати айді реєстрації іншого виборця й проголосувати до нього раніше. Звісно, що не можна відкидати варіант з шахраюванням з боку виборчої комісії.
5. Чи може хтось (інший виборець, ВК, стороння людина) таємно змінити голос в бюлетені?

   Таємно цього зробити не вдасться, оскільки публікується остаточний список бюлетенів та їх вміст після завершення процесу голосування.

Звісно, що не можна відкидати варіант з шахраюванням з боку виборчої комісії.

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

   Так, оскільки виводиться список бюлетенів та їх вміст після голосування. Звісно, що не можна відкидати варіант з шахраюванням з боку виборчої комісії.

Висновок:

У ході виконання лабораторної роботи було змодельовано та досліджено протокол електронного голосування з двома виборчими комісіями, де для кодування повідомлень використовувався метод Ель-Гамаля, а для реалізації електронного цифрового підпису – алгоритм DSA.

Під час дослідження було встановлено, що протокол ефективно запобігає голосуванню неавторизованих осіб завдяки попередній верифікації виборців реєстраційним бюро. Також виборці не можуть проголосувати кілька разів через перевірку статусу кожного виборця перед голосуванням. Протокол забезпечує певний рівень анонімності, оскільки бюлетені шифруються без прив’язки до особистості виборця. Однак можливе ідентифікування виборців через реєстраційні ID. Можливість проголосувати замість іншого виборця існує за умов доступу до ID реєстрації, але це малоймовірно без шахрайства з боку реєстраційного бюро або виборчої комісії. Зміна голосу після його відправки також малоймовірна, оскільки після завершення голосування публікуються бюлетені з їхнім вмістом, що забезпечує контроль за підсумками.

Таким чином, протокол забезпечує базові функції безпеки, верифікації виборців і шифрування голосів, однак не може повністю гарантувати абсолютну анонімність та захист від маніпуляцій.