ДОДАТОК А

Слайди презентації

Рисунок А.1 – Слайд №1

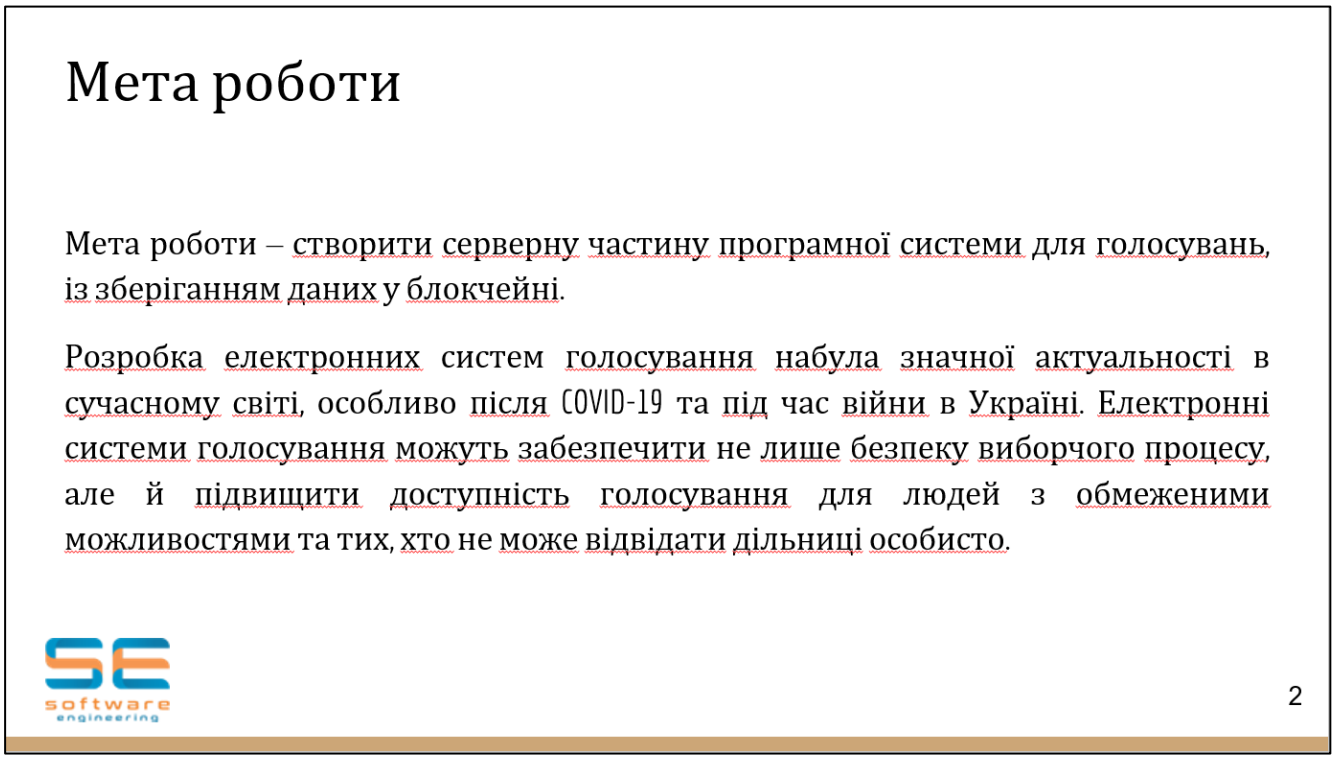


Рисунок А.2 – Слайд №2

Аналіз проблеми (аналіз існуючих рішень)

Проаналізовані конкуренти налічують: Horizone State, Vocdoni, Follow My Vote.

Наявні прогалини у конкурентів:

1. Horizone state – застій прогресу, помилки під час розробки, через що закрились під натиском юридичних росходів.
2. Vocdoni – обмежена масштабованість, централізація.
3. Follow My Vote – надає можливість зміни голосу, що створює вектори для атак.

3

Рисунок А.3 – Слайд №3

Постановка задачі та опис системи

Необхідно створити систему для голосувань, яка матиме достатній рівень безпеки а результати будуть відкритими. Користування системою має бути простим та доступним для звичайного користувача

Система має надавати повний функціонал для голосування та його життєвого циклу. Блокчейн частина буде лише перевіряти та зберігати свідчення про критично важливі дані задля зменшення можливих витрат на транзакції. Основна логіка та реалізація криптографічних протоколів відбувається на сервері.

4

Рисунок А.4 – Слайд №4

Вибір технологій розробки

Основний набір технологій розробки: мова програмування Go(Golang) та фреймворк Gin для обробки HTTP запитів. Мова програмування Python для сервісів з важкими криптографічними алгоритмами. Мова програмування Solidity для розробки смарт контрактів.

SE software engineering

5

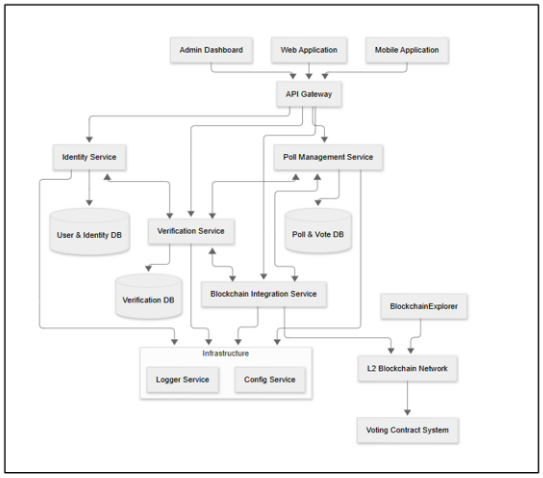
Рисунок А.5 – Слайд №5

Архітектура створенного програмного забезпечення

Гібридне поєднання Backend & Blockchain

Чотири основні сервіси: Identity, Poll Management, Verification, Blockchain Integration.
Identity, Poll Management та Verification мають власні бази даних.

Зі смарт контрактами взаємодіє лише Blockchain Integration Service.



SE software engineering

6

Рисунок А.6 – Слайд №6

Опис програмного забезпечення, що було використано у дослідженні

Програмну систему було розроблено у 2 етапи: реалізація основного функціоналу з відкритими голосуваннями, подальше доповнення методами для закритих голосувань.

Мова програмування Go була створена для мікросервісної архітектури, що значно спростило розробку. Багатий набір бібліотек Python для криптографії також значно пришвидшив реалізацію.

software engineering

7

Рисунок А.7 – Слайд №7

Дизайн системи

Було використано методологію Domain-Driven-Design(DDD). Кожен сервіс поділено на шари відповідно до Clean Architecture.

Сервіси було розроблено у послідовності Identity › Smart-contratcs › Poll Manager + Blockchain Integration › Verfication Service. Використовувалось Incremental Development – поступове додавання функціональності. Розвиток було спланово по фазах та етапах.

Для взаємодії сервісів із клієнтом та між собою використано фреймворк Gin, для мови Python - Fastapi. Також використано бібліотеки, які вже стали стандартом: zap для логування, viper для конфігурації системи, та інші.

software engineering

8

Рисунок А.8 – Слайд №8

Приклад реалізації

```go
// Service combines all service interfaces
type Service struct {
  Auth AuthService
  User UserService
  Role RoleService
  Key  KeyService
}
```

```go
// UserService defines methods for user management
type UserService interface {
  Create(ctx context.Context, input model.UserCreate) (*model.User, error)

  GetByID(ctx context.Context, id uint) (*model.User, error)
  GetByEmail(ctx context.Context, email string) (*model.User, error)
  Update(ctx context.Context, id uint, input model.UserUpdate) (*model.User, error)
  Delete(ctx context.Context, id uint) error
  List(ctx context.Context, offset, limit int) ([]model.User, error)
  ChangePassword(ctx context.Context, id uint, oldPassword, newPassword string) error
  GenerateKeyPair(ctx context.Context, userID uint) error
}
```

```go
// KeyService defines methods for cryptographic key operations
type KeyService interface {
  GenerateKeyPair() (publicKey, privateKey string, err error)
  EncryptPrivateKey(privateKey string) (string, error)
  DecryptPrivateKey(encryptedKey string) (string, error)
}
```

software engineering

9

Рисунок А.9 – Слайд №9

Приклад реалізації

```go
func (s *EncryptionService) GenerateECDSAKeyPair() (string, string, error) {
  privateKey, err := crypto.GenerateKey()
  if err != nil {
    return "", "", err
  }

  // Приватний ключ в PEM (custom блок)
  privateKeyBytes := crypto.FromECDSA(privateKey)
  privateKeyPEM := pem.EncodeToMemory(
    &pem.Block{
      Type:  "SECP256K1 PRIVATE KEY",
      Bytes: privateKeyBytes,
    },
  )

  publicKeyBytes := crypto.FromECDSAPub(&privateKey.PublicKey)
  publicKeyHex := hex.EncodeToString(publicKeyBytes[1:])

  return publicKeyHex, string(privateKeyPEM), nil
}
```

```go
func (bm *BlockchainMonitor) processPendingTransactions(ctx context.Context) (int, error) {
  pendingTxs, err := bm.repository.BlockchainTransaction.GetPendingTransactions(ctx)
  if err != nil {
    return 0, fmt.Errorf("getting pending transactions: %w", err)
  }

  if len(pendingTxs) == 0 {
    bm.logger.Println("No pending transactions found")
    return 0, nil
  }

  bm.logger.Printf("Processing %d pending transactions", len(pendingTxs))

  processed := 0
  for _, tx := range pendingTxs {
    if err := bm.checkTransactionStatus(ctx, &tx); err != nil {
      bm.logger.Printf("Error checking transaction %s: %v", safeString(tx.TxHash), err)
      continue
    }
    processed++
  }

  return processed, nil
}
```

software engineering

10

Рисунок А.10 – Слайд №10

Приклад реалізації

```go
// PublicKeyToAddress converts publi hex key into ethereum address
func PublicKeyToAddress(publicKeyHex string) (common.Address, error) {
  publicKeyHex = strings.TrimPrefix(publicKeyHex, "0x")

  if len(publicKeyHex) != 128 && len(publicKeyHex) != 130 {
    return common.Address{}, fmt.Errorf("invalid public key length: expected 128 or 130 characters, got %d", len(publicKeyHex))
  }
  if len(publicKeyHex) == 130 && publicKeyHex[:2] == "04" {
    publicKeyHex = publicKeyHex[2:]
  }

  publicKeyBytes, err := hex.DecodeString(publicKeyHex)
  if err != nil {
    return common.Address{}, fmt.Errorf("failed to decode public key hex: %w", err)
  }
  if len(publicKeyBytes) != 64 {
    return common.Address{}, fmt.Errorf("invalid public key bytes length: expected 64, got %d", len(publicKeyBytes))
  }

  pubKey := &ecdsa.PublicKey{
    Curve: crypto.S256(),
  }

  pubKey.X = new(big.Int).SetBytes(publicKeyBytes[:32])
  pubKey.Y = new(big.Int).SetBytes(publicKeyBytes[32:])

  if !pubKey.Curve.IsOnCurve(pubKey.X, pubKey.Y) {
    return common.Address{}, fmt.Errorf("public key point is not on the secp256k1 curve")
  }

  address := crypto.PubkeyToAddress(*pubKey)

  return address, nil
}
```

software engineering

11

Рисунок А.11 – Слайд №11

Тестування

```
✓ 23/23                                                    2.9s
✓ identity-api 510ms
  ✓ pkg/crypto 510ms
    ✓ encrypt_test.go 510ms
      ✓ TestNewEncryptionService 0.0ms
      ✓ TestEncryptDecrypt_Success 0.0ms
        ✓ empty_data
        ✓ long_data
        ✓ long_key_(truncated)
        ✓ normal_data_with_32-byte_key
        ✓ short_key_(padded)
        ✓ unicode_data
      ✓ TestEncrypt_ConsistentFormat 0.0ms
      ✓ TestDecrypt_InvalidInput 0.0ms
        ✓ corrupted_ciphertext
        ✓ empty_string
        ✓ invalid_base64
        ✓ too_short_ciphertext
      ✓ TestGenerateRSAKeyPair_Success 50ms
      ✓ TestGenerateRSAKeyPair_MultipleGenerations 450ms
      ✓ TestGenerateECDSAKeyPair_Success 10ms
      ✓ TestGenerateECDSAKeyPair_MultipleGenerations 0.0ms
      ✓ TestKeyHandling_EdgeCases 0.0ms
        ✓ empty_key
        ✓ exactly_32_bytes_key
        ✓ less_than_32_bytes_key
        ✓ more_than_32_bytes_key
```

```
goos: windows
goarch: amd64
pkg: identity-api/pkg/crypto
cpu: Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz
=== RUN   BenchmarkGenerateECDSAKeyPair
BenchmarkGenerateECDSAKeyPair
BenchmarkGenerateECDSAKeyPair-8
   34537             35142 ns/op            2448 B/op         21 allocs/op
PASS
ok      identity-api/pkg/crypto 1.799s
```

```
goos: windows
goarch: amd64
pkg: identity-api/pkg/crypto
cpu: Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz
=== RUN   BenchmarkEncrypt
BenchmarkEncrypt
BenchmarkEncrypt-8
 1343876               880.4 ns/op          1360 B/op          9 allocs/op
PASS
ok      identity-api/pkg/crypto 2.299s
```

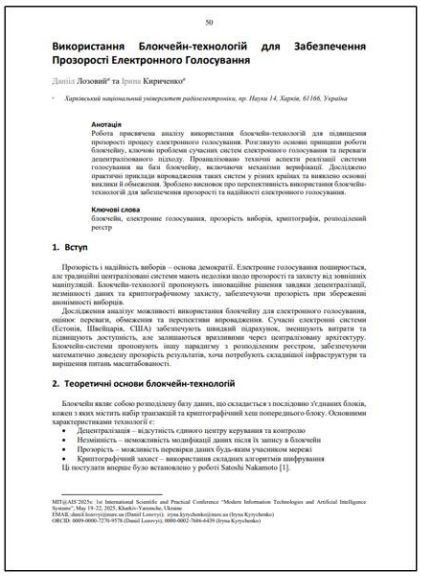
```
goos: windows
goarch: amd64
pkg: identity-api/pkg/crypto
cpu: Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz
=== RUN   BenchmarkDecrypt
BenchmarkDecrypt
BenchmarkDecrypt-8
 1877174               628.0 ns/op          1104 B/op          6 allocs/op
PASS
ok      identity-api/pkg/crypto 2.035s
```

software engineering

12

Рисунок А.12 – Слайд №12

Публікація результатів



50

Використання Блокчейн-технологій для Забезпечення Прозорості Електронного Голосування

Данііл Лозовий[a] та Ірина Кириченко[a]

[a] *Харківський національний університет радіоелектроніки, пр. Науки 14, Харків, 61166, Україна*

Анотація

Робота присвячена аналізу використання блокчейн-технологій для підвищення прозорості процесу електронного голосування. Розглянуто основні принципи роботи блокчейну, ключові проблеми сучасних систем електронного голосування та переваги децентралізованого підходу. Проаналізовано технічні аспекти реалізації системи голосування на базі блокчейну, включаючи механізми верифікації. Досліджено практичні приклади впровадження таких систем у різних країнах та виявлено основні виклики й обмеження. Зроблено висновок про перспективність використання блокчейн-технологій для забезпечення прозорості та надійності електронного голосування.

Ключові слова

блокчейн, електронне голосування, прозорість виборів, криптографія, розподілений реєстр

1. Вступ

Прозорість і надійність виборів – основа демократії. Електронне голосування поширюється, але традиційні централізовані системи мають недоліки щодо прозорості та захисту від зовнішніх маніпуляцій. Блокчейн-технології пропонують інноваційне рішення завдяки децентралізації, незмінності даних та криптографічному захисту, забезпечуючи прозорість при збереженні анонімності виборців.

Дослідження аналізує можливості використання блокчейну для електронного голосування, оцінює переваги, обмеження та перспективи впровадження. Сучасні електронні системи (Естонія, Швейцарія, США) забезпечують швидкий підрахунок, зменшують витрати та підвищують доступність, але залишаються вразливими через централізовану архітектуру. Блокчейн-системи пропонують іншу парадигму з розподіленим реєстром, забезпечуючи математично доведену прозорість результатів, хоча потребують складної інфраструктури та вирішення питань масштабованості.

2. Теоретичні основи блокчейн-технологій

Блокчейн являє собою розподілену базу даних, що складається з послідовно зʼєднаних блоків, кожен з яких містить набір транзакцій та криптографічний хеш попереднього блоку. Основними характеристиками технології є:

- Децентралізація – відсутність єдиного центру керування та контролю
- Незмінність – неможливість модифікації даних після їх запису в блокчейн
- Прозорість – можливість перевірки даних будь-яким учасником мережі
- Криптографічний захист – використання складних алгоритмів шифрування

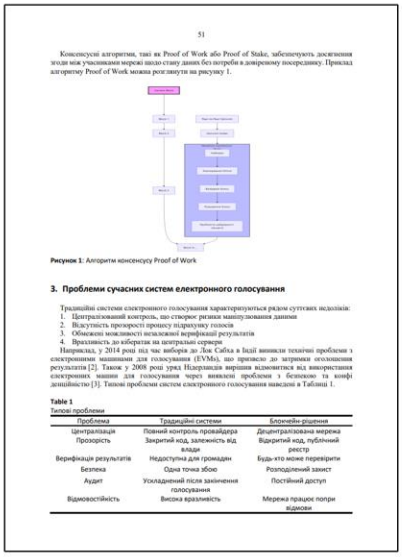
Ці постулати вперше було встановлено у роботі Satoshi Nakamoto [1].

MIT@AIS'2025: 1st International Scientific and Practical Conference "Modern Information Technologies and Artificial Intelligence Systems", May 19-22, 2025, Kharkiv-Yaremche, Ukraine
EMAIL: daniil.lozovyi@nure.ua (Daniil Lozovyi); iryna.kyrychenko@nure.ua (Iryna Kyrychenko)
ORCID: 0009-0000-7270-9578 (Daniil Lozovyi); 0000-0002-7686-6439 (Iryna Kyrychenko)

software engineering

Рисунок А.13 – слайд №13

Публікація результатів



51

Консенсусні алгоритми, такі як Proof of Work або Proof of Stake, забезпечують досягнення згоди між учасниками мережі щодо стану даних без потреби в довіреному посереднику. Приклад алгоритму Proof of Work можна розглянути на рисунку 1.

Рисунок 1: Алгоритм консенсусу Proof of Work

3. Проблеми сучасних систем електронного голосування

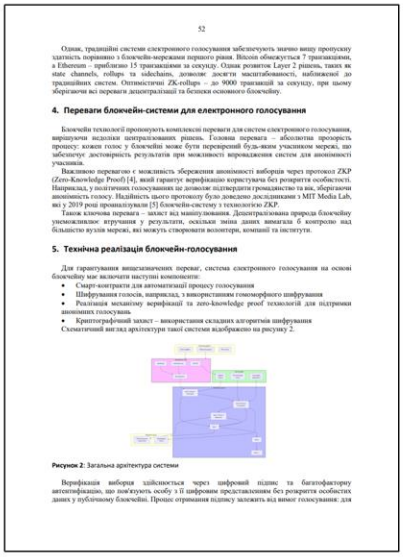
Традиційні системи електронного голосування характеризуються рядом суттєвих недоліків:

1. Централізований контроль, що створює ризики маніпулювання даними
2. Відсутність прозорості процесу підрахунку голосів
3. Обмежені можливості незалежної верифікації результатів
4. Вразливість до кібератак на центральні сервери

Наприклад, у 2014 році під час виборів до Лок Сабха в Індії виникли технічні проблеми з електронними машинами для голосування (EVMs), що призвело до затримки оголошення результатів [2]. Також у 2008 році уряд Нідерландів вирішив відмовитися від використання електронних машин для голосування через виявлені проблеми з безпекою та конфіденційністю [3]. Типові проблеми систем електронного голосування наведені в Таблиці 1.

Table 1
Типові проблеми

Проблема	Традиційні системи	Блокчейн-рішення
Централізація	Повний контроль провайдера	Децентралізована мережа
Прозорість	Закритий код, залежність від влади	Відкритий код, публічний реєстр
Верифікація результатів	Недоступна для громадян	Будь-хто може перевірити
Безпека	Одна точка збою	Розподілений захист
Аудит	Ускладнений після закінчення голосування	Постійний доступ
Відмовостійкість	Висока вразливість	Мережа працює попри відмови



52

Однак, традиційні системи електронного голосування забезпечують значно вищу пропускну здатність порівняно з блокчейн-мережами першого рівня. Bitcoin обмежується 7 транзакціями, а Ethereum – приблизно 15 транзакціями за секунду. Однак розвиток Layer 2 рішень, таких як state channels, rollups та sidechains, дозволяє досягти масштабованості, наближеної до традиційних систем. Оптимістичні ZK-rollups – до 9000 транзакцій за секунду, при цьому зберігаючи всі переваги децентралізації та безпеки основного блокчейну.

4. Переваги блокчейн-системи для електронного голосування

Блокчейн технології пропонують комплексні переваги для систем електронного голосування, вирішуючи недоліки централізованих рішень. Головна перевага – абсолютна прозорість процесу: кожен голос у блокчейні може бути перевірений будь-яким учасником мережі, що забезпечує достовірність результатів при можливості впровадження систем для анонімності учасників.

Важливою перевагою є можливість збереження анонімності виборців через протокол ZKP (Zero-Knowledge Proof) [4], який гарантує верифікацію користувачів без розкриття особистості. Наприклад, у політичних голосуваннях це дозволяє підтвердити громадянство та вік, зберігаючи анонімність голосу. Надійність цього протоколу було доведено дослідниками з MIT Media Lab, які у 2019 році промоделювали [5] блокчейн-систему з технологією ZKP.

Також ключова перевага – захист від маніпулювання. Децентралізована природа блокчейну унеможливлює втручання у результати, оскільки зміна даних вимагала б контролю над більшістю вузлів мережі, які можуть створювати волонтери, компанії та інститути.

5. Технічна реалізація блокчейн-голосування

Для гарантування вищезазначених переваг, система електронного голосування на основі блокчейну має включати наступні компоненти:

- Смарт-контракти для автоматизації процесу голосування
- Шифрування голосів, наприклад, з використанням гомоморфного шифрування
- Реалізація механізму верифікації та zero-knowledge proof технологій для підтримки анонімних голосувань
- Криптографічний захист – використання складних алгоритмів шифрування

Схематичний вигляд архітектури такої системи відображено на рисунку 2.

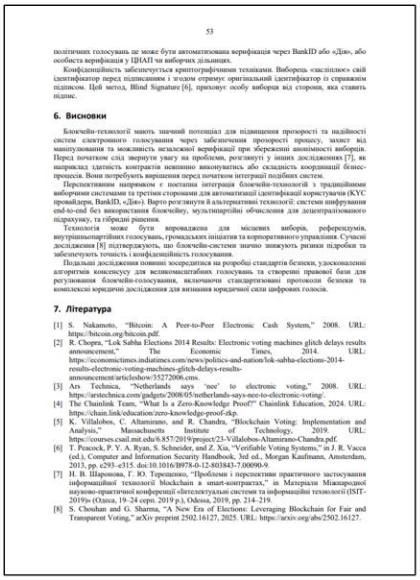
Рисунок 2: Загальна архітектура системи

Верифікація виборця здійснюється через цифровий підпис та багатофакторну автентифікацію, що пов'язують особу з її цифровим представленням без розкриття особистих даних у публічному блокчейні. Процес отримання підпису залежить від вимог голосування: для

software engineering

Рисунок А.14 – слайд №14

Публікація результатів



53

політичних голосувань це може бути автоматизована верифікація через BankID або «Дія», або особиста верифікація у ЦНАП чи виборчих дільницях.

Конфіденційність забезпечується криптографічними техніками. Виборець «засліпило» свій ідентифікатор перед підписанням і згодом отримує оригінальний ідентифікатор із справжнім підписом. Цей метод, Blind Signature [6], приховує особу виборця від сторони, яка ставить підпис.

6. Висновки

Блокчейн-технології мають значний потенціал для підвищення прозорості та надійності систем електронного голосування через забезпечення прозорості процесу, захист від маніпулювання та можливість незалежної верифікації при збереженні анонімності виборців. Перед початком слід звернути увагу на проблеми, розглянуті у інших дослідженнях [7], як наприклад здатність контрактів невпинно виконуватись або складність координації бізнес-процесів. Вони потребують вирішення перед початком інтеграції подібних систем.

Перспективним напрямком є поетапна інтеграція блокчейн-технологій з традиційними виборчими системами та третіми сторонами для автоматизації ідентифікації користувачів (KYC провайдери, BankID, «Дія»). Варто розглянути й альтернативні технології: системи шифрування end-to-end без використання блокчейну, мультипартійні обчислення для децентралізованого підрахунку, та гібридні рішення.

Технологія може бути впроваджена для місцевих виборів, референдумів, внутрішньопартійних голосувань, громадських ініціатив та корпоративного управління. Сучасні дослідження [8] підтверджують, що блокчейн-системи значно зменшують ризики підробки та забезпечують точність і конфіденційність голосування.

Подальші дослідження повинні зосередитися на розробці стандартів безпеки, удосконаленні алгоритмів консенсусу для великомасштабних голосувань та створенні правової бази для регулювання блокчейн-голосування, включаючи стандартизовані протоколи безпеки та комплексні юридичні дослідження для визнання юридичної сили цифрових голосів.

7. Література

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. URL: https://bitcoin.org/bitcoin.pdf.

[2] R. Chopra, "Lok Sabha Elections 2014 Results: Electronic voting machines glitch delays results announcement," The Economic Times, 2014. URL: https://economictimes.indiatimes.com/news/politics-and-nation/lok-sabha-elections-2014-results-electronic-voting-machines-glitch-delays-results-announcement/articleshow/35272006.cms.

[3] Ars Technica, "Netherlands says 'nee' to electronic voting," 2008. URL: https://arstechnica.com/gadgets/2008/05/netherlands-says-nee-to-electronic-voting/.

[4] The Chainlink Team, "What Is a Zero-Knowledge Proof?" Chainlink Education, 2024. URL: https://chain.link/education/zero-knowledge-proof-zkp.

[5] K. Villalobos, C. Altamirano, and R. Chandra, "Blockchain Voting: Implementation and Analysis," Massachusetts Institute of Technology, 2019. URL: https://courses.csail.mit.edu/6.857/2019/project/23-Villalobos-Altamirano-Chandra.pdf.

[6] T. Peacock, P. Y. A. Ryan, S. Schneider, and Z. Xia, "Verifiable Voting Systems," in J. R. Vacca (ed.), Computer and Information Security Handbook, 3rd ed., Morgan Kaufmann, Amsterdam, 2013, pp. e293–e315. doi:10.1016/B978-0-12-803843-7.00090-9.

[7] Н. В. Шаронова, Г. Ю. Терещенко, "Проблеми і перспективи практичного застосування інформаційної технології blockchain в smart-контрактах," in Матеріали Міжнародної науково-практичної конференції «Інтелектуальні системи та інформаційні технології (ISIT-2019)» (Одеса, 19–24 серп. 2019 р.). Одеса, 2019. pp. 214–219.

[8] S. Chouhan and G. Sharma, "A New Era of Elections: Leveraging Blockchain for Fair and Transparent Voting," arXiv preprint 2502.16127, 2025. URL: https://arxiv.org/abs/2502.16127.

Рисунок А.15 – слайд №15

Підсумки

За результатами работи отримано систему, достатньо наближену до теоретичного ідеалу та необхідної реалізації. Ключові проблеми: спрощена оптимізація газу, спрощена реалізація криптографічних доказів.

Систему можна використовувати для невеликих голосувань, в тому числі у закритих спільнотах.

Подальший розвиток передбачає якісну оптимізацію для зменшення витрат, повну власну реалізацію криптографічних доказів, покарщення функціоналу.

Рисунок А.16 – слайд №16

ДОДАТОК Б

Специфікація програмного продукту

1 ВСТУП

1.1 Огляд проекту

У часи пандемії та війни, коли проведення масових заходів, таких як голосування, стають небезпечними або взагалі неможливими використання онлайн рішень виглядає закономірним розвитком. А впровадження блокчейну у подібні системи підвищує надійність та відсоток довіри серед виборців.

1.2 Мета

Створити серверну частину онлайн платформи для голосувань, із зберіганням даних у блокчейні.

1.3 Область застосування

Серверну частину цього продукту можна запустити на будь-якій операційні системі, де встановлено Docker, для роботи контейнерів із сервісами додатку. Смарт контракти можуть бути розміщені у глобальній мережі Polygon, тож ніяк не залежать від операційної системи чи специфікацій серверу У випадку, якщо буде використовуватись локальна блокчейн мережа, як, наприклад, Hardhat, вона також може бути створена всередині контейнеру Docker.

1.4 Короткий зміст

В першу чергу цільова аудиторія буде звертати увагу на:

1. Зручний інтерфейс веб-додатку;
2. Швидкість роботи додатку;
3. Загальний рівень безпеки;
4. Захищеність персональних даних.

1.5 Означення та абревіатури

АПІ (англ. API, Application Programming Interface) – це набір правил та протоколів, що дозволяє різним програмним системам взаємодіяти між собою через обмін даними та викликами функцій.

Блокчейн (англ. Blockchain) – це розподілена база даних, що складається з ланцюжка криптографічно зв'язаних блоків, які зберігають незмінні записи транзакцій та забезпечують децентралізоване зберігання інформації без необхідності довіреної третьої сторони.

2 ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

Розроблювана програма система надасть користувачам можливість проводити власні голосування та брати у них участь онлайн, з будь-якої точки світу. У системі будуть дотримані достатні рівні безпеки для успішного голосування.

2.2 Функції продукту

Функціями продукту є реєстрація/вхід, створення голосувань, глобальний пошук по ним, фільтрація, голосування, автоматичне закінчення та публікація результатів у блокчейн.

2.3 Характеристики користувачів

Цільовим користувачем виступає будь-який користувач із доступом до інтернету та необхідністю провести голосування або взяти у ньому участь. Знання блокчейн технологій та криптографія не потребуються.

2.4 Загальні обмеження

2.4.1 Операційне середовище

Середа розробки серверної частини: JetBrains Goland 2024.3.5, MS Visual Studio Code.

В якості СКБД було використано PostgreSQL, а саме pgAdmin 4.

2.4.2 Технологія розробки

Golang, Solidity, Python.

2.5 Припущення й залежності

Припущення:

1. Система голосувань буде мати попит серед користувачів, оскільки надає можливості створення голосувань різних типів під різні потреби

Залежності:

1. Початкова версія додатку матиме підтримку лише голосувань виду один з обраних варіантів;
2. Блокчейн частина додатку буде розміщена у Hardhat мережі, із можливістю розгортання у Polygon мережі;
3. Критично важливі операції виконуються на серверній частині для зменшення витрат на транзакції, їх результат зберігається у блокчейні для збереження цілісності даних.

3 КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до інтерфейсів

3.1.1 Апаратний інтерфейс

Аппаратний інтерфейс блокчейн-платформи для голосувань реалізується через веб-додаток та мобільний застосунок, які взаємодіють з мікросервісною архітектурою бекенду через API з аутентифікацією. Уся складність приховується від коритсувача.

3.1.2 Комунікаційний протокол

HTTP/HTTPS, Web3-JSON-RPC

3.1.3 Вимоги до пам’яті

Вимогами до пам’яті є наступними:

1. Windows 10, 11 (64 бітні), Windows Server 2016+:
    - 4 GB RAM;
    - 2 GB дискового простору для роботи БД;
2. MacOS 10.15 Catalina, 11 Big Sur, 12 Monterey, 13 Ventura (64-bit):
    - 4 GB RAM;
    - 2 GB дискового простору для роботи БД;
3. GNU/Linux (kernel 2.6.32+, 64-bit):
    - 3 GB RAM;
    - 1.5 GB дискового простору для роботи БД.

Розподіл пам'яті по компонентах GNU/Linux може відрізнятися залежно від дистрибутива. Ubuntu 20.04+, Debian 11+, CentOS 8+ мають оптимізовані вимоги завдяки ефективнішому управлінню контейнерами.

3.2 Атребути програмного продукту

3.2.1 Безпека

Усі критичні дані користувача, такі як паролі та приватні ключі, шифруються. Перевірка доказів із нульовим розголошенням відбувається на стороні серверної частини, результат перевірки надсилається у блокчейн.

3.2.2 Супроводжуваність

Після випуску програмного продукту планується супроводжуваність, розвиток функціоналу, подальше покращення безпеки.

3.2.3 Переносимість

Так як це програмний продукт це серверний додаток, вимагається щоб він працював на усіх серверах які відповідають технічним вимогам та вимогам до пам’яті.

3.2.4 Продуктивність

Вимоги до продуктивності не висуваються

3.2.5 Вимоги до бази даних

Підтримуються версії сервера PostgreSQL 13.x та старші.

ДОДАТОК В

Фрагменти коду програмної реалізації

1. Реалізація протоколу ZKP у вигляді псевдокоду

```
    // Генерація доказу користувачем
    function генерувати_ZKP_доказ(
        сліпий_підпис,           // Підпис від довіреної особи
        відкритий_ключ_підписанта, // Публічний ключ довіреної особи
        голосування_id,           // Ідентифікатор голосування
        варіанти_голосу,          // Вибрані варіанти відповіді
        приватний_ключ_користувача  //  Приватний  ключ  користувача  для
створення доказу
    ):
        // Створюємо публічні входи (будуть доступні верифікатору)
        публічні_входи = {
            голосування_id,
            хеш(варіанти_голосу),
            відкритий_ключ_підписанта
        }

        // Створюємо приватні входи (відомі лише доказнику)
        приватні_входи = {
            сліпий_підпис,
            розсліплений_підпис,
            приватний_ключ_користувача
        }

        // Генеруємо доказ за допомогою бібліотеки ZKP
        доказ = zk_snark_генерувати(
            схема_доказу,     // Скомпільована схема доказу
            публічні_входи,
            приватні_входи
        )

        return (доказ, публічні_входи)

    // Верифікація доказу смарт-контрактом
    function перевірити_ZKP_доказ(доказ, публічні_входи):
        // Перевіряємо, що голосування активне
        if не_є_активним(публічні_входи.голосування_id):
            return false

        // Перевіряємо, що цей доказ ще не використовувався
        if є_використаним_доказом(доказ):
            return false

        // Перевіряємо зк-доказ
        результат = zk_snark_верифікувати(
            схема_верифікації,  // Параметри верифікації
            публічні_входи,
            доказ
        )

        if результат == true:
```

```
            // Позначаємо доказ як використаний
            позначити_доказ_використаним(доказ)
            // Зберігаємо голос
            зберегти_голос(публічні_входи.голосування_id,
публічні_входи.хеш_варіантів)

        return результат
```

ДОДАТОК Г

Рисунки цікавих фрагментів коду смарт-контракту

```
function storeResults(
        uint256 _pollId,
        uint256[] memory _optionCounts
    ) external onlyOwner whenNotPaused nonReentrant {
        require(_pollId > 0, "Invalid poll ID");
        require(!resultsExist[_pollId], "Results already stored");
        require(
            _optionCounts.length > 0 &&
                _optionCounts.length ≤ MAX_OPTIONS_STORAGE,
            "Invalid options count"
        );

        // Перевіряємо що голосування існує
        require(pollRegistry.pollExists(_pollId), "Poll does not exist");

        // ПОтримуємо метаданні голосування
        PollRegistry.PollMetadata memory poll = pollRegistry.getPoll(_pollId);

        // Перевіряємо що голосування завершено
        require(
            poll.status == PollRegistry.PollStatus.Ended ||
                poll.status == PollRegistry.PollStatus.Failed,
            "Poll is not finished"
        );
```

Рисунок Г.1 – Перевірки умов перед збереженням результатів

```
// Підраховуємо загальну кількість голосів та знаходимо переможця
        uint256 totalVotes = 0;
        uint256 winnerOptionId = 1;
        uint256 maxVotes = _optionCounts[0];

        for (uint256 i = 0; i < _optionCounts.length; i++) {
            totalVotes += _optionCounts[i];

            // Знаходимо переможну опцію
            if (_optionCounts[i] > maxVotes) {
                maxVotes = _optionCounts[i];
                winnerOptionId = i + 1; // optionId починається з 1
            }
        }

        // Перевіряємо що загальна кількість голосів співпадає з VoteVerifier
        uint256 verifierVoteCount = voteVerifier.getPollVoteCount(_pollId);
        require(
            totalVotes == verifierVoteCount,
            "Vote count mismatch with VoteVerifier"
        );

        // Створюємо хеш результатів
        bytes32 resultsHash = keccak256(
            abi.encodePacked(
                _pollId,
                totalVotes,
                winnerOptionId,
                _optionCounts,
                block.timestamp,
                msg.sender
            )
        );
```

Рисунок Г.2 – Незалежний від бекенду підрахунок переможця

```
function verifyResults(uint256 _pollId) external view returns (bool) {
    if (!resultsExist[_pollId]) {
        return false;
    }

    PollResult memory result = pollResults[_pollId];

    // Перевіряємо загальну кількість голосів
    uint256 verifierTotalVotes = voteVerifier.getPollVoteCount(_pollId);
    if (result.totalVotes != verifierTotalVotes) {
        return false;
    }

    // Перевіряємо голоси за кожну опцію
    for (uint256 i = 0; i < result.optionCounts.length; i++) {
        uint256 optionId = i + 1;
        uint256 verifierOptionVotes = voteVerifier.getOptionVoteCount(
            _pollId,
            optionId
        );
        if (result.optionCounts[i] != verifierOptionVotes) {
            return false;
        }
    }

    return true;
}
```

Рисунок Г.3 – Верифікація результату

ДОДАТОК Д

Тези І міжнародної науково-технічної конференції «Сучасні інформаційні технології та системи штучного інтелекту» MIT@AIS-2025

Використання Блокчейн-технологій для Забезпечення Прозорості Електронного Голосування

Даніїл Лозовий[a] та Ірина Кириченко[a]

[a] *Харківський національний університет радіоелектроніки, пр. Науки 14, Харків, 61166, Україна*

Анотація

Робота присвячена аналізу використання блокчейн-технологій для підвищення прозорості процесу електронного голосування. Розглянуто основні принципи роботи блокчейну, ключові проблеми сучасних систем електронного голосування та переваги децентралізованого підходу. Проаналізовано технічні аспекти реалізації системи голосування на базі блокчейну, включаючи механізми верифікації. Досліджено практичні приклади впровадження таких систем у різних країнах та виявлено основні виклики й обмеження. Зроблено висновок про перспективність використання блокчейн-технологій для забезпечення прозорості та надійності електронного голосування.

Ключові слова

блокчейн, електронне голосування, прозорість виборів, криптографія, розподілений реєстр

1. Вступ

Прозорість і надійність виборів – основа демократії. Електронне голосування поширюється, але традиційні централізовані системи мають недоліки щодо прозорості та захисту від зовнішніх маніпуляцій. Блокчейн-технології пропонують інноваційне рішення завдяки децентралізації, незмінності даних та криптографічному захисту, забезпечуючи прозорість при збереженні анонімності виборців.

Дослідження аналізує можливості використання блокчейну для електронного голосування, оцінює переваги, обмеження та перспективи впровадження. Сучасні електронні системи (Естонія, Швейцарія, США) забезпечують швидкий підрахунок, зменшують витрати та підвищують доступність, але залишаються вразливими через централізовану архітектуру. Блокчейн-системи пропонують іншу парадигму з розподіленим реєстром, забезпечуючи математично доведену прозорість результатів, хоча потребують складнішої інфраструктури та вирішення питань масштабованості.

2. Теоретичні основи блокчейн-технологій

Блокчейн являє собою розподілену базу даних, що складається з послідовно з'єднаних блоків, кожен з яких містить набір транзакцій та криптографічний хеш попереднього блоку. Основними характеристиками технології є:

- Децентралізація – відсутність єдиного центру керування та контролю
- Незмінність – неможливість модифікації даних після їх запису в блокчейн
- Прозорість – можливість перевірки даних будь-яким учасником мережі
- Криптографічний захист – використання складних алгоритмів шифрування

Ці постулати вперше було встановлено у роботі Satoshi Nakamoto [1].

MIT@AIS'2025s: 1st International Scientific and Practical Conference "Modern Information Technologies and Artificial Intelligence Systems", May 19–22, 2025, Kharkiv-Yaremche, Ukraine
EMAIL:daniil.lozovyi@nure.ua (Daniil Lozovyi); iryna.kyrychenko@nure.ua (Iryna Kyrychenko)
ORCID: 0009-0000-7270-9578 (Daniil Lozovyi); 0000-0002-7686-6439 (Iryna Kyrychenko)

Консенсусні алгоритми, такі як Proof of Work або Proof of Stake, забезпечують досягнення згоди між учасниками мережі щодо стану даних без потреби в довіреному посереднику. Приклад алгоритму Proof of Work можна розглянути на рисунку 1.

Рисунок 1: Алгоритм консенсусу Proof of Work

3. Проблеми сучасних систем електронного голосування

Традиційні системи електронного голосування характеризуються рядом суттєвих недоліків:

1. Централізований контроль, що створює ризики маніпулювання даними
2. Відсутність прозорості процесу підрахунку голосів
3. Обмежені можливості незалежної верифікації результатів
4. Вразливість до кібератак на центральні сервери

Наприклад, у 2014 році під час виборів до Лок Сабха в Індії виникли технічні проблеми з електронними машинами для голосування (EVMs), що призвело до затримки оголошення результатів [2]. Також у 2008 році уряд Нідерландів вирішив відмовитися від використання електронних машин для голосування через виявлені проблеми з безпекою та конфіденційністю [3]. Типові проблеми систем електронного голосування наведені в Таблиці 1.

Table 1
Типові проблеми

Проблема	Традиційні системи	Блокчейн-рішення
Централізація	Повний контроль провайдера	Децентралізована мережа
Прозорість	Закритий код, залежність від влади	Відкритий код, публічний реєстр
Верифікація результатів	Недоступна для громадян	Будь-хто може перевірити
Безпека	Одна точка збою	Розподілений захист
Аудит	Ускладнений після закінчення голосування	Постійний доступ
Відмовостійкість	Висока вразливість	Мережа працює попри відмови

Однак, традиційні системи електронного голосування забезпечують значно вищу пропускну здатність порівняно з блокчейн-мережами першого рівня. Bitcoin обмежується 7 транзакціями, а Ethereum – приблизно 15 транзакціями за секунду. Однак розвиток Layer 2 рішень, таких як state channels, rollups та sidechains, дозволяє досягти масштабованості, наближеної до традиційних систем. Оптимістичні ZK-rollups – до 9000 транзакцій за секунду, при цьому зберігаючи всі переваги децентралізації та безпеки основного блокчейну.

4. Переваги блокчейн-системи для електронного голосування

Блокчейн технології пропонують комплексні переваги для систем електронного голосування, вирішуючи недоліки централізованих рішень. Головна перевага – абсолютна прозорість процесу: кожен голос у блокчейні може бути перевірений будь-яким учасником мережі, що забезпечує достовірність результатів при можливості впровадження систем для анонімності учасників.

Важливою перевагою є можливість збереження анонімності виборців через протокол ZKP (Zero-Knowledge Proof) [4], який гарантує верифікацію користувача без розкриття особистості. Наприклад, у політичних голосуваннях це дозволяє підтвердити громадянство та вік, зберігаючи анонімність голосу. Надійність цього протоколу було доведено дослідниками з MIT Media Lab, які у 2019 році проаналізували [5] блокчейн-систему з технологією ZKP.

Також ключова перевага – захист від маніпулювання. Децентралізована природа блокчейну унеможливлює втручання у результати, оскільки зміна даних вимагала б контролю над більшістю вузлів мережі, які можуть створювати волонтери, компанії та інститути.

5. Технічна реалізація блокчейн-голосування

Для гарантування вищезазначених переваг, система електронного голосування на основі блокчейну має включати наступні компоненти:

- Смарт-контракти для автоматизації процесу голосування
- Шифрування голосів, наприклад, з використанням гомоморфного шифрування
- Реалізація механізму верифікації та zero-knowledge proof технологій для підтримки анонімних голосувань
- Криптографічний захист – використання складних алгоритмів шифрування

Схематичний вигляд архітектури такої системи відображено на рисунку 2.

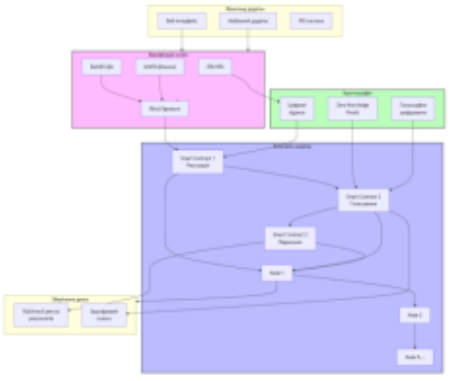


Рисунок 2: Загальна архітектура системи

Верифікація виборця здійснюється через цифровий підпис та багатофакторну автентифікацію, що пов'язують особу з її цифровим представленням без розкриття особистих даних у публічному блокчейні. Процес отримання підпису залежить від вимог голосування: для

політичних голосувань це може бути автоматизована верифікація через BankID або «Дія», або особиста верифікація у ЦНАП чи виборчих дільницях.

Конфіденційність забезпечується криптографічними техніками. Виборець «засліплює» свій ідентифікатор перед підписанням і згодом отримує оригінальний ідентифікатор із справжнім підписом. Цей метод, Blind Signature [6], приховує особу виборця від сторони, яка ставить підпис.

6. Висновки

Блокчейн-технології мають значний потенціал для підвищення прозорості та надійності систем електронного голосування через забезпечення прозорості процесу, захист від маніпулювання та можливість незалежної верифікації при збереженні анонімності виборців. Перед початком слід звернути увагу на проблеми, розглянуті у інших дослідженнях [7], як наприклад здатність контрактів невпинно виконуватись або складність координації бізнес-процесів. Вони потребують вирішення перед початком інтеграції подібних систем.

Перспективним напрямком є поетапна інтеграція блокчейн-технологій з традиційними виборчими системами та третіми сторонами для автоматизації ідентифікації користувачів (KYC провайдери, BankID, «Дія»). Варто розглянути й альтернативні технології: системи шифрування end-to-end без використання блокчейну, мультипартійні обчислення для децентралізованого підрахунку, та гібридні рішення.

Технологія може бути впроваджена для місцевих виборів, референдумів, внутрішньопартійних голосувань, громадських ініціатив та корпоративного управління. Сучасні дослідження [8] підтверджують, що блокчейн-системи значно знижують ризики підробки та забезпечують точність і конфіденційність голосування.

Подальші дослідження повинні зосередитися на розробці стандартів безпеки, удосконаленні алгоритмів консенсусу для великомасштабних голосувань та створенні правової бази для регулювання блокчейн-голосування, включаючи стандартизовані протоколи безпеки та комплексні юридичні дослідження для визнання юридичної сили цифрових голосів.

7. Література

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008. URL: https://bitcoin.org/bitcoin.pdf.

[2] R. Chopra, “Lok Sabha Elections 2014 Results: Electronic voting machines glitch delays results announcement,” The Economic Times, 2014. URL: https://economictimes.indiatimes.com/news/politics-and-nation/lok-sabha-elections-2014-results-electronic-voting-machines-glitch-delays-results-announcement/articleshow/35272006.cms.

[3] Ars Technica, “Netherlands says ‘nee’ to electronic voting,” 2008. URL: https://arstechnica.com/gadgets/2008/05/netherlands-says-nee-to-electronic-voting/.

[4] The Chainlink Team, “What Is a Zero-Knowledge Proof?” Chainlink Education, 2024. URL: https://chain.link/education/zero-knowledge-proof-zkp.

[5] K. Villalobos, C. Altamirano, and R. Chandra, “Blockchain Voting: Implementation and Analysis,” Massachusetts Institute of Technology, 2019. URL: https://courses.csail.mit.edu/6.857/2019/project/23-Villalobos-Altamirano-Chandra.pdf.

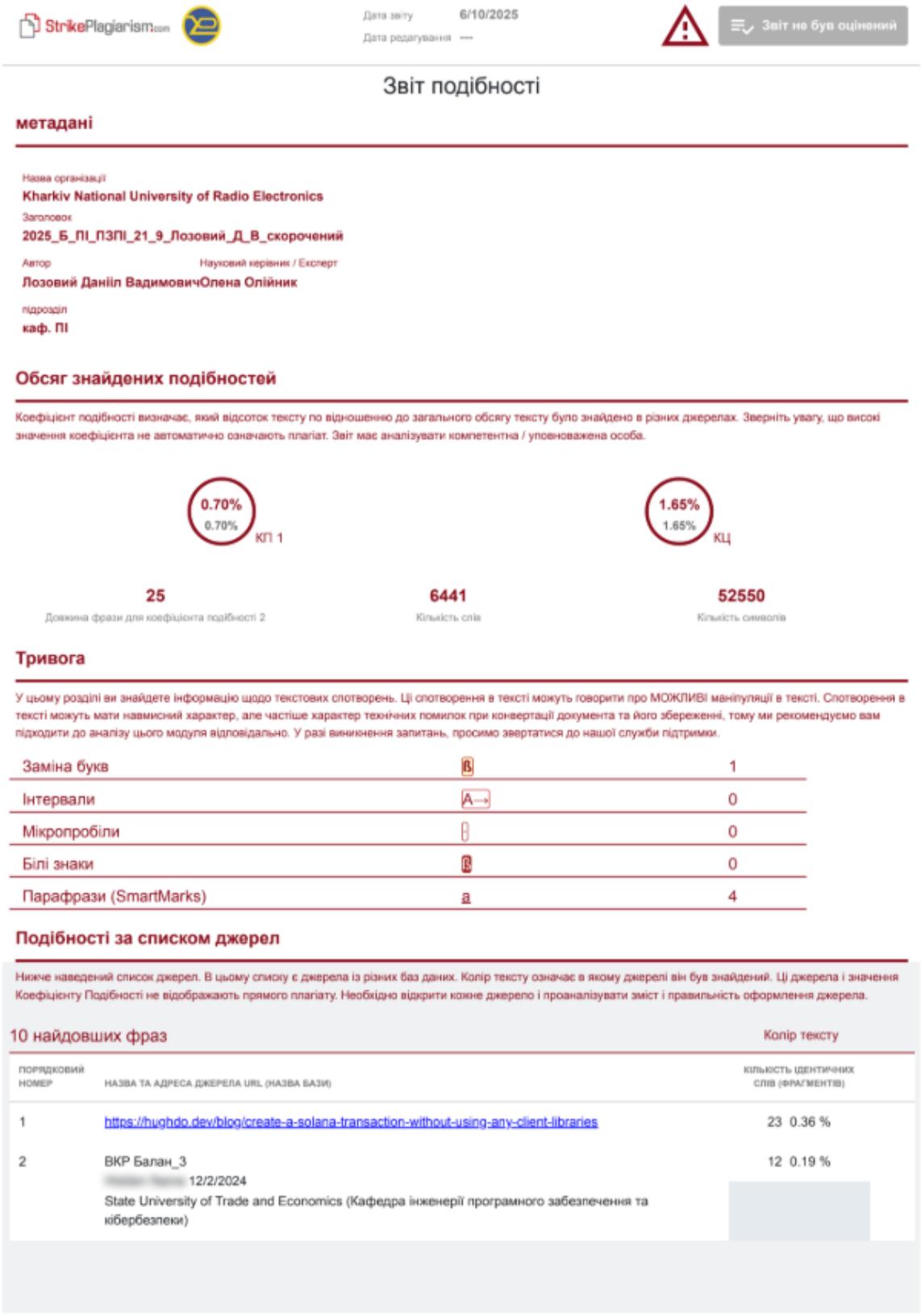
[6] T. Peacock, P. Y. A. Ryan, S. Schneider, and Z. Xia, “Verifiable Voting Systems,” in J. R. Vacca (ed.), Computer and Information Security Handbook, 3rd ed., Morgan Kaufmann, Amsterdam, 2013, pp. e293–e315. doi:10.1016/B978-0-12-803843-7.00090-9.

[7] Н. В. Шаронова, Г. Ю. Терещенко, “Проблеми і перспективи практичного застосування інформаційної технології blockchain в smart-контрактах,” in Матеріали Міжнародної науково-практичної конференції «Інтелектуальні системи та інформаційні технології (ISIT-2019)» (Одеса, 19–24 серп. 2019 р.), Odessa, 2019, pp. 214–219.

[8] S. Chouhan and G. Sharma, “A New Era of Elections: Leveraging Blockchain for Fair and Transparent Voting,” arXiv preprint 2502.16127, 2025. URL: https://arxiv.org/abs/2502.16127.

ДОДАТОК Е

Звіт з результатами перевірки на унікальність тексту в базі ХНУРЕ



StrikePlagiarism.com

Дата звіту 6/10/2025
Дата редагування ---

Звіт не був оцінений

Звіт подібності

метадані

Назва організації
Kharkiv National University of Radio Electronics
Заголовок
2025_Б_ПІ_ПЗПІ_21_9_Лозовий_Д_В_скорочений
Автор Науковий керівник / Експерт
Лозовий Даніїл ВадимовичОлена Олійник
підрозділ
каф. ПІ

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

0.70% 0.70% КП 1

1.65% 1.65% КЦ

25	6441	52550
Довжина фрази для коефіцієнта подібності 2	Кількість слів	Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		1
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		4

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз — Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://hughdo.dev/blog/create-a-solana-transaction-without-using-any-client-libraries	23 0.36 %
2	ВКР Балан_3 [redacted] 12/2/2024 State University of Trade and Economics (Кафедра інженерії програмного забезпечення та кібербезпеки)	12 0.19 %