КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І АРХІТЕКТУРИ

Факультет автоматизації та інформаційних технологій

Кафедра кібербезпеки та комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему:

Розробка децентралізованої системи управління цифровою ідентичністю для IoT-пристроїв (DID for IoT)

Здобувач: Мочарська Валерія Тарасівна

Керівник: Гуменний Д.О.

Київ – 2026

Актуальність, мета, об’єкт, предмет

- **Актуальність:** IoT-пристрої не мають вбудованої надійної ідентифікації; PKI має єдину точку відмови (CA), високу вартість, потребує синхронної перевірки; відсутність контролю власника
- **Мета:** розробити прототип децентралізованої системи управління цифровою ідентичністю для IoT на базі DID (W3C), блокчейну Polygon та IPFS
- **Об’єкт:** процеси автентифікації, авторизації та управління життєвим циклом ідентичностей у розподілених мережах IoT
- **Предмет:** методи, архітектурні рішення, криптографічні протоколи та програмні засоби для децентралізованого управління цифровою ідентичністю IoT-пристроїв на основі стандартів W3C DID

Проблеми централізованих моделей ідентифікації

Проблеми PKI / OAuth2 / SAML для IoT

- Єдина точка відмови (CA)
- Великі обсяги даних
- Час верифікації
- Час перевірки статусу
- Високе гавантаження на процесор
- Відсутність контролю власника

Критерій	Централізовані	Федеративні
Єдина точка відмови	Так	Так
Розмір токена	SAML-асершн: 2-8 КБ, X.509: ~200-220 Б	SAML: ті ж 2-8 КБ
Час верифікації	OIDC: 1-2 мс SAML: 18-25 мс	XML-верифікація: до 20+ мс
Час перевірки статусу	OCSP: до 5 с таймаут, кеш 24 год CRL: ~2 МБ, OCSP: ~430 мс	Синхронне звернення до IdP
CPU навантаження	OAuth2 для 1000 пристроїв: до 500 мс затримки	shibd: 99% CPU, час відповіді з 50 мс до >1 хв

Вибір DID-методу та криптостеку

DID-метод: did:polygon (Amoy Testnet)

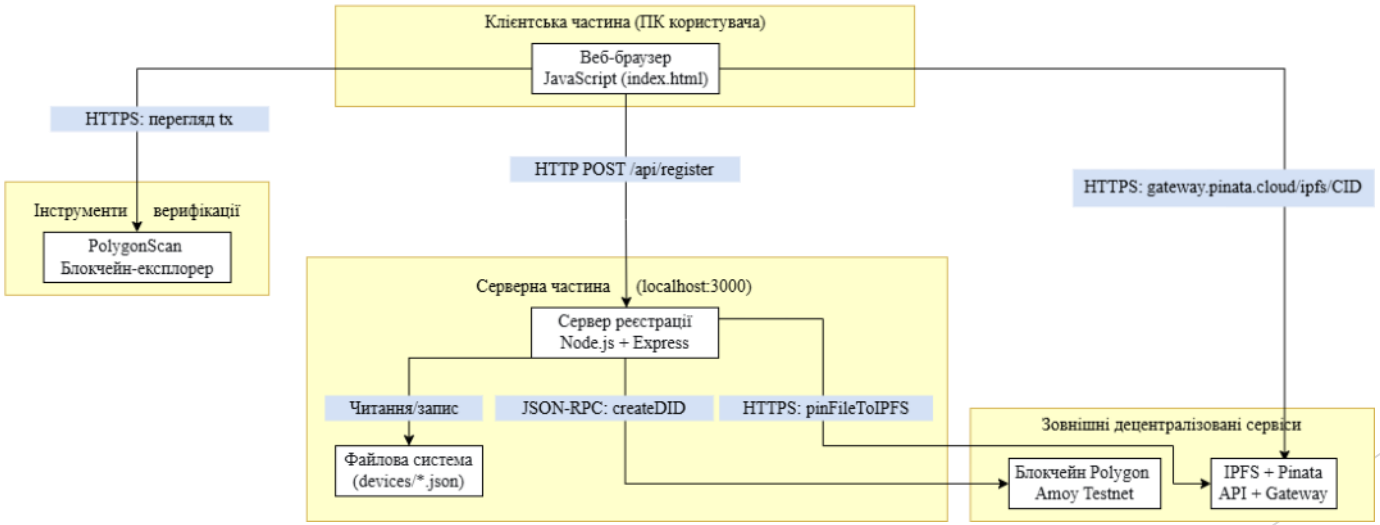
- Безкоштовна реєстрація (Faucet)
- Підтримка ротації ключів
- Сумісність з MetaMask

Криптографічний стек (Lightweight Crypto):

- Підпис: Ed25519 (32 Б, 2 КБ RAM)
- Шифрування: потоковий ChaCha20-Poly1305
- Хеш: SHA-256

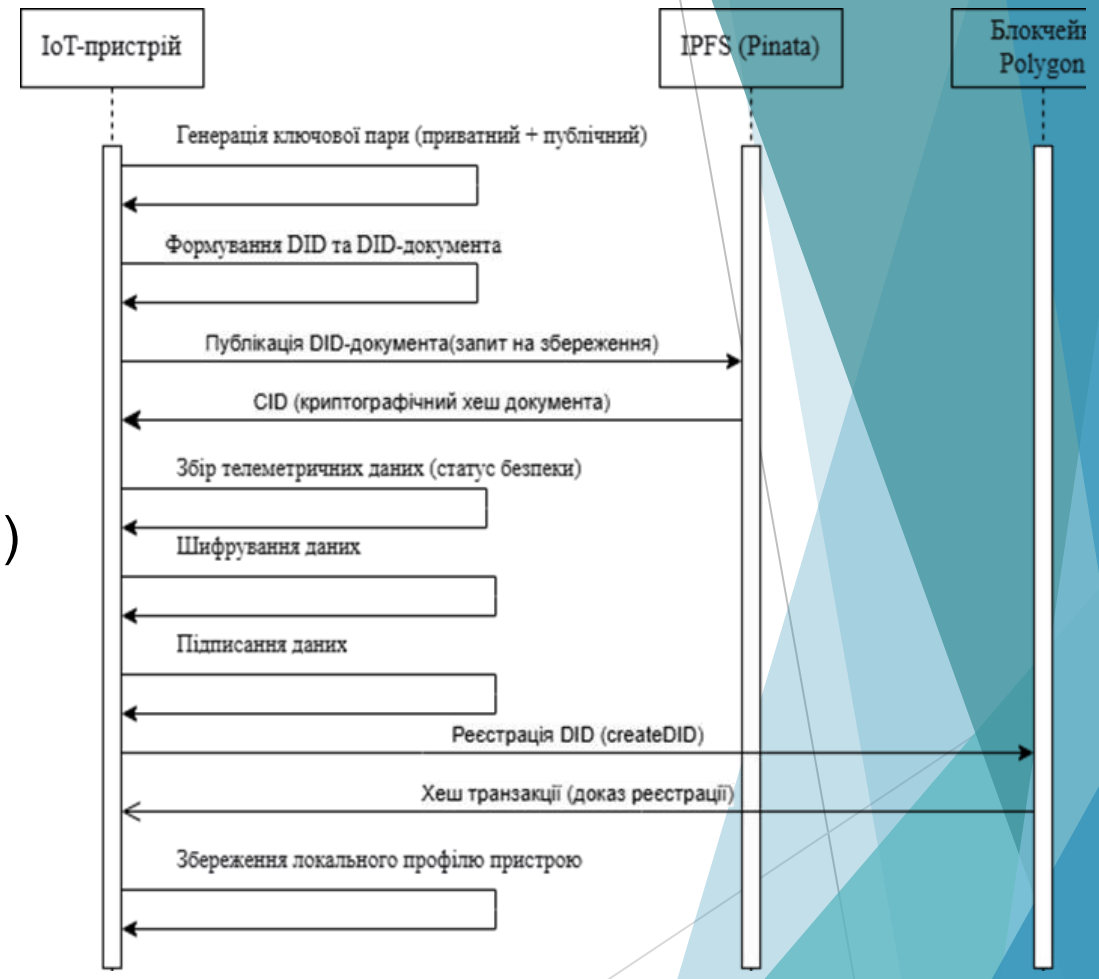
Архітектура системи (компоненти та взаємодія)

- IoT-пристрій (Node.js): генерація ключів, DID, шифрування, підпис, реєстрація
- Блокчейн Polygon Amoy Testnet: реєстр DID, смарт-контракт createDID, registrationTxHash
- IPFS (Pinata): сховище DID-документів, CID = криптографічний хеш вмісту
- Гарантії: автентичність (Ed25519), цілісність (CID/хеш), незмінність (блокчейн), конфіденційність (ChaCha20)



Протокол реєстрації IoT-пристрою

- Генерація пари Ed25519
- Формування DID та DID-документа (JSON-LD, W3C)
- Публікація в IPFS → CID
- Шифрування (ChaCha20) + підпис (Ed25519)
- Реєстрація в блокчейні → TxHash (виклик createDID(owner, didString))
- Збереження локального JSON-профілю



Програмна реалізація криптографічних операцій

- **Шифрування**: encryptData() / decryptData() – ChaCha20-Poly1305, IV = 12 байт
- **Генерація ключів**: ethers.Wallet.createRandom() – Ed25519
- **Підпис**: wallet.signMessage()
- **Верифікація**: ethers.verifyMessage() – відновлює адресу підписувача
- **Хешування**: crypto.createHash('sha256')

```
65  function encryptData(data, key) {
66      const iv = crypto.randomBytes(12);
67      const cipher = crypto.createCipheriv('chacha20-poly1305', key, iv);
68      let encrypted = cipher.update(data, 'utf8', 'hex');
69      encrypted += cipher.final('hex');
70      const authTag = cipher.getAuthTag();
71      return { encrypted, iv: iv.toString('hex'), authTag: authTag.toString('hex') };
72  }
```

```
const deviceWallet = ethers.Wallet.createRandom();
const deviceAddress = deviceWallet.address;
const devicePrivateKey = deviceWallet.privateKey;
const signature = await deviceWallet.signMessage(readingsJson);
```

```
async function verifySignature(deviceData) {
    const recoveredAddress = ethers.verifyMessage(deviceData.signedMessage, deviceData.signature);
    const isValid = (recoveredAddress.toLowerCase() === deviceData.address.toLowerCase());
```

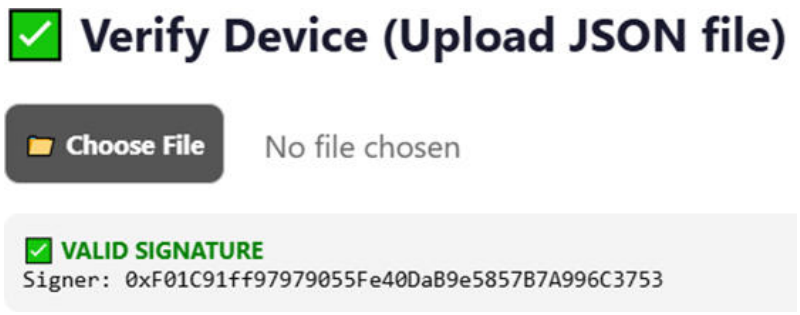
Експериментальне тестування – результати

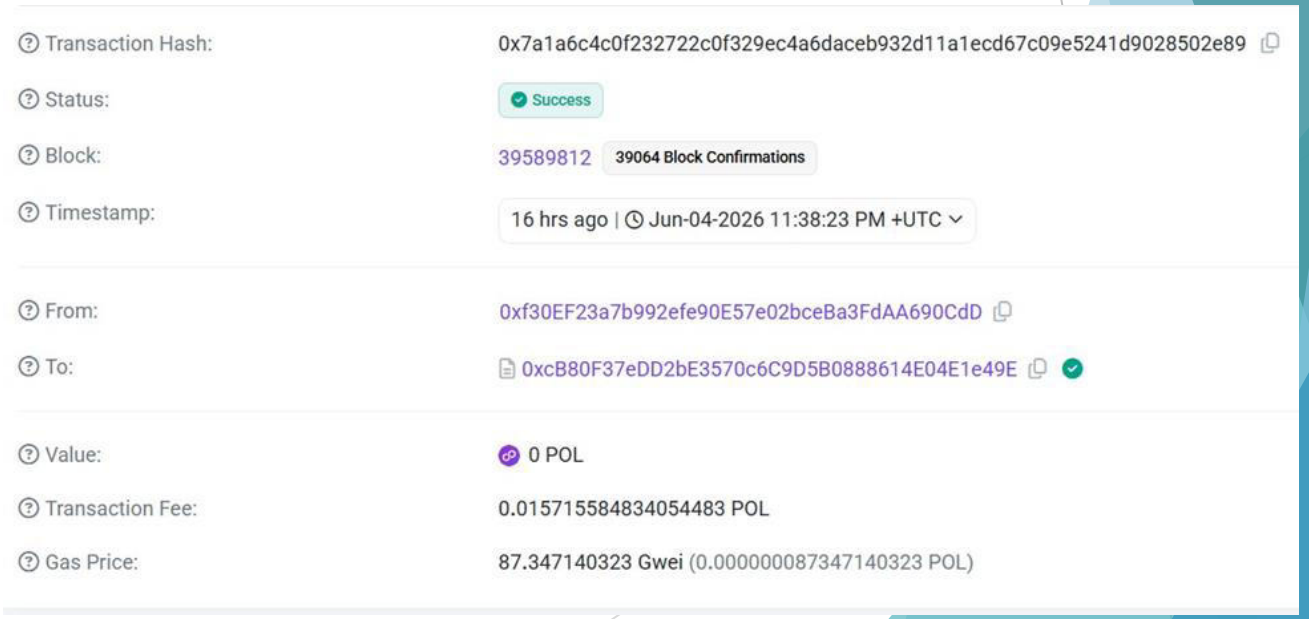
- Умови: AMD Ryzen 5 5600H (3.3 ГГц), 16 ГБ DDR4, Windows 10, Node.js v24.00.0, 20 вимірювань
- Результати (пристрої r1–r4):
- Генерація ключів: 11.44–156.49 мс (FR1 ≤100 мс) - 3/4
- Підписання: 0.82–3.46 мс (FR2 ≤10 мс) - 4/4
- Шифрування: 0.21–14.13 мс (FR3 ≤1 мс) - 2/4
- DID-документ: 505 байт (FR4 ≤500 байт) - перевищено на 1%
- Верифікація: 2.84–7.65 мс (очікувано ~60 мс)

Operation	Requirement	Result	Status
Key Generation (Ed25519)	≤ 100 ms	**43.92 ms**	✅ PASS
Message Signing (Ed25519)	≤ 10 ms	**3.46 ms**	✅ PASS
Encryption ChaCha20 (1 KB)	≤ 1 ms	**14.128 ms**	❌ FAIL
DID Document Size	≤ 500 bytes	**505 bytes**	❌ FAIL
Signature Verification	~60 ms	**3.21 ms**	✅ PASS

Публічна верифікація та порівняння з PKI

- Публічна верифікація (без центрального сервера): DID → IPFS (CID) → публічний ключ → ethers.verifyMessage()
- Перевірка в блокчейні: PolygonScan показує TxHash, статус, блок, час, виклик createDID
- Переваги над PKI: немає CA / офлайн-верифікація / нульова вартість / контроль власника





Висновки та результати роботи

- Розроблено архітектуру DID for IoT (Polygon + IPFS + Node.js)
- Обґрунтовано та реалізовано криптостек: Ed25519, ChaCha20, SHA-256
- Створено програмний прототип: реєстрація, шифрування, підпис, верифікація
- Експериментально підтверджено FR1-FR4
- Доведено переваги над PKI (SPOF, вартість, офлайн, контроль власника)