УДК 621.395.7 (043.2)

В. В. Гуменюк[1]
К. В. Безпалий[1]

СИСТЕМА ЕЛЕКТРОННОГО ГОЛОСУВАННЯ З ПІДВИЩЕНОЮ ДОСТОВІРНІСТЮ

[1] Вінницький національний технічний університет

Анотація

В роботі описано розробку децентралізованої системи електронного голосування з використанням смарт-контрактів на платформі Ethereum. Особливу увагу приділено забезпеченню конфіденційності введених даних та запобіганню їх несанкціонованому доступу. Для підвищення достовірності електронного голосування в блокчейн-мережі були впроваджені механізми захисту на основі смарт-контрактів. Поставлена мета досягнута шляхом створення смарт-контракту на блокчейн-платформі Ethereum, інтеграції системи голосування з мережею блокчейн з можливістю обробки даних у довіреному середовищі виконання (Trusted Execution Environment), та налаштування взаємодії між мережею і системою голосування. Конфіденційність даних забезпечено криптографічними методами.

Ключові слова: кіберзагрози, інформаційна безпека, децентралізована система електронного голосування, смарт-контракти, Ethereum, блокчейн-мережа.

Abstract

The study describes the development of a decentralized electronic voting system using smart contracts on the Ethereum platform. Special attention is given to ensuring the confidentiality of the entered data and preventing unauthorized access. To enhance the reliability of electronic voting in the blockchain network, security mechanisms based on smart contracts were implemented. The goal was achieved by creating a smart contract on the Ethereum blockchain platform, integrating the voting system with the blockchain network with the capability to process data in a Trusted Execution Environment, and setting up interaction between the network and the voting system. Data confidentiality is ensured through cryptographic methods.

Keywords: *cyber threats, information security, decentralized electronic voting system, smart contracts, Ethereum, blockchain network.*

Вступ

Голосування є ключовим методом прийняття рішень у групах, що включає підрахунок голосів кожного виборця для визначення колективної думки. Існують конфіденційне та відкрите голосування. У відкритому голосуванні всі знають, хто і як голосував, що забезпечує прозорість, але може призвести до тиску або підкупу. Конфіденційне голосування захищає інформацію про вибір кожного від третіх осіб, знижуючи ризики підкупу чи тиску, але у масштабних виборах може бути важко перевірити, що створює можливості для маніпуляцій. Електронне голосування використовує спеціальні електронні засоби для голосування, підрахунку голосів та оголошення результатів. Однією з його форм є Інтернет-голосування, яке скорочує час на підрахунок голосів і дозволяє виборцям голосувати дистанційно, зменшуючи витрати часу та коштів. Електронне голосування, засноване на криптографії, забезпечує анонімність і автоматичний підрахунок результатів, роблячи систему економічною, прозорою та неупередженою. Питання безпеки залишається ключовим, тому було розроблено низку протоколів, включаючи Mix networks, сліпий підпис, гомоморфне шифрування, кільцевий підпис та нульові докази. Інтернет-голосування стало ще надійнішим і ефективнішим з появою криптовалют та вдосконаленням шифрування, що підвищило довіру виборців до результатів голосування.

Проектування системи електронного голосування з підвищеною достовірністю на основі смарт-контрактів

Технологія blockchain, завдяки своїй децентралізованій природі, реплікації бази даних між учасниками, незмінності даних та збереженню усіх транзакцій у вигляді ланцюжка блоків, має значний потенціал для використання в електронних системах голосування. Ці характеристики дозволяють усунути основні недоліки традиційних електронних систем голосування, забезпечуючи високу ступінь безпеки та довіри до процесу голосування. Основні принципи побудови децентралізованих мереж і технології blockchain, а також вимоги до голосування із застосуванням blockchain вимагають підвищення достовірності системи електронного голосування з використанням смарт-контрактів для захисту конфіденційності. Для досягнення цієї мети необхідно виконати такі завдання: побудувати смарт-контракт на блокчейн-мережі Ethereum, інтегрувати систему голосування з можливістю обробки даних у довіреному середовищі виконання (Trusted Execution Environment), налаштувати зв'язок між мережею та системою голосування із забезпеченням конфіденційності даних криптографічними методами, та розробити графічний інтерфейс для взаємодії із системою голосування.

На даний момент існує значна кількість реалізацій блокчейнів, таких як Bitcoin, Ethereum, zCash, Waves, Ripple, Cardano, Stellar, IOTA тощо. Серед них, особливо Ethereum та Waves, мають на меті створення платформ для вирішення різних завдань за допомогою смарт-контрактів, які забезпечують різноманітні маніпуляції з даними у блокчейні. Хоча кожна з цих платформ має свої переваги, найефективнішою для розробки децентралізованих додатків наразі є Ethereum, оскільки інші платформи є відносно новими, мають менше навчальних матеріалів, а засоби для розробки на цих платформах часто змінюються, що ускладнює та робить менш ефективною реалізацію децентралізованих додатків.

Можливість виконувати різні маніпуляції з даними в ланцюжку блоків забезпечується спеціальним комп'ютерним алгоритмом, який називається смарт-контракт. Смарт-контракт задає правила зберігання даних та описує набори функцій для операцій з ними. Усі ці маніпуляції здійснюються за допомогою інтерфейсу, який створюється з вихідного коду окремо від компіляції і надає можливість виконувати двійковий код. Як і передбачається для блокчейну, всі дані в Ethereum надаються кожному учаснику мережі, оскільки вони реплікуються та розподіляються між користувачами. Зміни вносяться у вигляді транзакцій. У Ethereum конструкція транзакції включає: одержувача, електронний підпис відправника, суму переказу у валюті "ETH" (токен, що використовується в системі), коментар, ліміт "палива" на транзакцію (gasLimit) та вартість одиниці "палива" (gasPrice).

При взаємодії з Ethereum Dapp, користувач має можливість звертатися безпосередньо до блокчейну за допомогою спеціалізованого програмного забезпечення або через командний рядок на своєму пристрої. Для створення смарт-контрактів на платформі Ethereum найчастіше використовується об'єктно-орієнтована мова програмування Solidity, яка спеціалізується на предметній області. Вона була представлена ще у 2014 році і рекомендується до використання командою проекту Ethereum. В офіційному репозиторії Ethereum доступні документація та посібники для цієї мови програмування.

Система голосування надає такі основні функції: створення опитування з можливістю адміністративного контролю та реєстрації виборців, голосування з внесенням власних даних за обраний об'єкт, перегляд результатів голосування та перевірка пов'язаних зі смарт-контрактом транзакцій для забезпечення їх чесності (рис. 1). При зверненні до смарт-контракту користувач може створити опитування, передавши масив з елементами, такими як назва опитування, питання та варіанти голосування. Після створення опитування він може реєструвати нових користувачів, які зможуть голосувати.

Смарт-контракт також дозволяє отримати доступ до блоку з даними про результати голосування та переглядати історію транзакцій, завдяки можливостям блокчейн-мережі, у якій розгорнутий смарт-контракт. Логічне моделювання передбачає перевірку функціонування логічної схеми додатка без повної реалізації на даному етапі розробки. Цей процес дозволяє перевірити як логічні функції додатка, так і його тимчасові співвідношення. Для здійснення моделювання необхідно побудувати діаграму послідовності та діаграму варіантів використання програми, що допомагає виявити та виправити потенційні помилки на ранніх стадіях розробки.

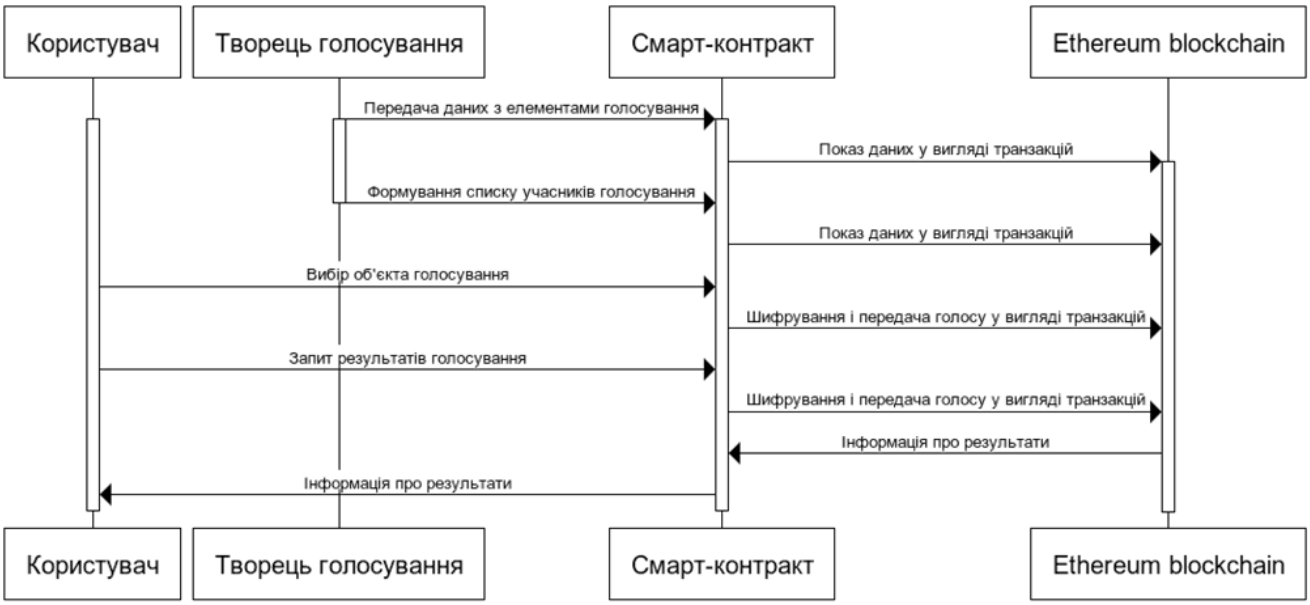

Рисунок 1. Діаграма роботи системи електронного голосування

Спочатку необхідно визначити базові вимоги до смарт-контракту, а саме які операції з даними в блокчейні він повинен виконувати. Виходячи з цих вимог, слід написати програмний код смарт-контракту. На початковому етапі у смарт-контракті мають бути реалізовані основні функції: конструктор, який отримуватиме масив об'єктів для голосування, надання автору опитування статусу адміністратора, функція, що дозволяє голосувати за обраний об'єкт, шифрування вибраного імені об'єкта для забезпечення конфіденційності, функція, яка повертає кількість голосів за кожний об'єкт, та функція, що передає дані у захищене середовище. У розроблюваній системі голосування на початку контракту використовується структура як комплексний тип даних, до яких будуть звертатися основні функції.

Intel SGX була розроблена для забезпечення безпечних віддалених обчислень, тобто для запуску програмного забезпечення на віддаленому комп'ютері, що потенційно належить ненадійній стороні, з гарантіями цілісності та конфіденційності даних. Intel SGX забезпечує захист від апаратних та програмних атак шляхом заборони зовнішнього читання з анклаву та запису до нього, незалежно від рівня привілеїв та режиму центрального процесора. Налагодження експлуатованих анклавів недоступне, але розробник може дозволити налагодження під час процесу розробки. Спілкування з анклавом здійснюється через спеціальні функції ECALL (Enclave Call) та OCALL (Outside Call), що ізолює анклав від основної ОС. Пам'ять анклаву зашифрована з використанням алгоритмів шифрування з відкритим ключем, ключі шифрування змінюються випадково при включенні живлення або перезавантаженні системи та зберігаються в ізольованому сховищі процесора. Доступ до даних анклаву можливий лише за допомогою коду, що використовує цей анклав.

Основний чейнкод мережі включає смарт-контракт PDnSC, який відповідає за прийняття персональних даних від peers, їх збереження у форматі JSON у базі даних як ключ-значення, встановлення загального секретного ключа з анклавом SGX, перевірку хеша анклаву, шифрування персональних даних за допомогою відкритого ключа, відправку даних у захищене середовище, розшифровку та запис результатів обробки до бази даних, з якої peers можуть отримати результати у відкритому вигляді.

Під час роботи програми з Intel SGX процес включає кілька ключових етапів. Спочатку програму поділяють на довірену і ненадійну частини. Анклав створюється у захищеному компоненті, і виконання коду передається в анклав через довірений виклик функції з ненадійної частини програми. Код анклаву обробляє дані у відкритому вигляді, при цьому доступ до цих даних із привілейованого рівня ОС заборонений. Після завершення обробки результати передаються назад у ненадійну частину програми,

яка продовжує виконання у звичайному режимі. Застосування довірених обчислювальних ресурсів поза мережею дозволяє збільшити пропускну спроможність і покращити конфіденційність даних. Використання Trusted Execution Environment забезпечує цілісність виконання програмного коду і гарантує дотримання вимог конфіденційності, що особливо важливо для обробки чутливих даних.

Висновки

Проаналізовано найбільш поширені платформи blockchain і обрано найбільш розвинену для побудови системи електронного голосування. Підвищено достовірність системи електронного голосування, для цього спочатку визначено мову програмування і реалізовано смарт-контракт, розвинуто механізм Trusted Execution Environment на прикладі Intel SGX з використанням особливостей реалізації додатків із застосуванням анклавів. Також Intel SGX використано для створення довіреного середовища обробки персональних даних при голосуванні завдяки ефективному механізму «сліпої» обробки конфіденційних даних. Використання анклавів забезпечило безпечне передавання персональних даних користувачів у довірене середовище.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Robert Watson, Trusted Code Base in a UNIX Environment [Електронний ресурс]. – Режим доступу: https://lists.freebsd.org/pipermail/trustedbsddiscuss/2000-April/000050.html

2. Thomas Knauth, Michael Steiner Integrating Intel SGX Remote Attestation with Transport Layer Security [Електронний ресурс]. – Режим доступу: https://arxiv.org/pdf/1801.05863.pdf

3. Rui Yuan, Hai-Bo Chen ShadowEth: Private Smart Contract on Public Blockchain [Електронний ресурс]. – Режим доступу: https://www.trustkernel.com/uploads/ pubs/shadoweth.pdf

4. Blockchain Security Solutions [Електронний ресурс]. – Режим доступу: https://safenet.gemalto.com/blockchain

5. Gisolfi D., Patel M., Radulovich R. Decentralized Identity Introduction. 2018 [Електронний ресурс]. – Режим доступу: https://www.ibm.com/downloads/cas/OPEQYEL

6. Dominik Meißner, Felix Engelmann PeQES: A Platform for Privacy-enhanced Quantitative Empirical Studies [Електронний ресурс]. – Режим доступу: https://arxiv.org/pdf/2103.05544.pdf

Гуменюк В'ячеслав Володимирович – асистент кафедри менеджменту та безпеки інформаційних систем, Вінницький національний технічний університет, Вінниця, e-mail: hvv@vntu.edu.ua

Безпалий Кирило Валерійович – асистент кафедри менеджменту та безпеки інформаційних систем, Вінницький національний технічний університет, Вінниця, e-mail: kyrylo.bezpalyi@vntu.edu.ua

Humeniuk Vyacheslav Volodymyrovych – associate assistant of the Department of Management and Security of Information Systems, Vinnytsia National Technical University, Vinnytsia, e-mail: hvv@vntu.edu.ua

Bezpalyi Kyrylo Valeriovitch – associate assistant of the Department of Management and Security of Information Systems, Vinnytsia National Technical University, Vinnytsia, e-mail: kyrylo.bezpalyi@vntu.edu.ua