КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (28), 2025
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2025.28.817
УДК 004.056

Пархоменко Іван Іванович
кандидат технічних наук, доцент, завідуючий кафедри КБЗІ
Факультет інформаційних технологій
Київський національний університет ім. Тараса Шевченка, Київ, Україна
ORCID ID: 0000-0001-6889-9284
ivan.parkhomenko@knu.ua

Огієвич Роман Віталійович
аспірант кафедри КБЗІ
Факультет інформаційних технологій
Київський національний університет ім. Тараса Шевченка, Київ, Україна
ORCID ID: 0009-0003-7948-1125
ohiievychr@fit.knu.ua

КРИПТОЕКОНОМІЧНА СТІЙКІСТЬ ДЕЦЕНТРАЛІЗОВАНОЇ МЕРЕЖІ З ВИКОРИСТАННЯМ ВИПАДКОВИХ ЧАСОВИХ ЧЕЛЕНДЖ-ТОКЕНІВ (RTCT)

Анотація. У статті розглядається підхід до підвищення криптоекономічної стійкості децентралізованих мереж за допомогою механізму випадкових часових челендж-токенів (Random Time Challenge Tokens, RTCT). Проаналізовано сучасні консенсусні механізми (Proof-of-Work, Proof-of-Stake, Proof-of-Burn) та їхні можливості протидії атакам типу 51% і атакам Сивілли. Показано, що класичні підходи забезпечують безпеку мережі створенням значного економічного бар'єра для зловмисників — наприклад, майнери змушені інвестувати в обладнання та електроенергію, що підвищує вартість 51%-атаки до надто високого рівня. Водночас уразливості залишаються: нападники можуть орендувати обчислювальні ресурси або скористатися недостатньо активними валідаторами. Запропонований механізм RTCT передбачає генерування випадкових криптографічних викликів (челенджів) у випадкові моменти часу, на які вузли мережі повинні надати криптографічно доведений відгук (челендж-токен) з подальшим його спаленням. Такий процес створює непередбачуване навантаження та постійні витрати для учасників мережі, що ускладнює реалізацію атак шляхом значного зростання їх вартості. У роботі математично формалізовано процес RTCT та оцінено залежність вартості атаки від параметра складності m такого челенджу. Отримані результати підтверджують, що зі збільшенням складності m криптоекономічна атака потребує експоненційно більших витрат, що робить мережу більш стійкою. Обговорено переваги та потенційні недоліки підходу RTCT, а також можливі напрямки подальших досліджень, зокрема оптимізацію частоти викликів та інтеграцію RTCT з існуючими протоколами консенсусу.

Ключові слова: криптоекономічна стійкість; децентралізована мережа; челендж-токен; доказ роботи; доказ ставки; доказ спалення; атака 51%; атака Сивілли; спам-атака.

ВСТУП

У сучасних децентралізованих мережах, таких як блокчейн-системи, безпека та надійність досягаються поєднанням криптографічних методів і економічних стимулів. Криптоекономічна стійкість означає здатність мережі протистояти зловмисним діям завдяки економічно невигідним умовам для атакуючих. Найбільш відомими прикладами є протоколи консенсусу: Proof-of-Work (PoW), де учасники (майнери) витрачають обчислювальні ресурси та електроенергію, і Proof-of-Stake (PoS), де учасники (валідатори) блокують власні монети як заставу. Обидва механізми створюють фінансові

перешкоди для атак. Так, у PoW-мережах на кшталт Bitcoin для реалізації 51%-атаки необхідно зосередити понад половину глобальної обчислювальної потужності, що потребує гігантських витрат на спеціалізоване обладнання та електроенергію [1], [12]. У PoS-мережах зловмиснику потрібно накопичити понад 51% (або 2/3) всіх монет, поставлених на стейкінг, що у великих мережах становить десятки мільярдів доларів, і при спробі атаки ця застава буде втрачена через механізм слешингу (штрафного знищення монет). Таким чином, базові консенсусні моделі забезпечують безпеку блокчейнів за рахунок надзвичайно високої вартості успішної атаки.

Утім, з розвитком технологій та появою нових загроз виникає потреба посилити стійкість мережі додатковими механізмами. По-перше, існують атаки типу Сивілли, коли один суб'єкт створює багато фальшивих вузлів, аби вплинути на роботу мережі. По-друге, можливе тимчасове захоплення контролю над мережею шляхом оренди ресурсів (наприклад, хешрейту через сервіси на кшталт NiceHash) або змови валідаторів, що не потребує довготривалого інвестування у власне обладнання. По-третє, в деяких сценаріях зловмисник може чекати оптимального моменту для атаки через вразливості в розкладі генерації блоків чи низької активності учасників. Таким чином, актуальною є задача розробки додаткового захисного шару, який би робив спроби атак ще більш економічно невигідними та важкопередбачуваними.

Саме таким підходом є використання випадкових часових челендж-токенів (RTCT). Ідея полягає в тому, щоб мережа періодично й непередбачувано генерувала виклики, які потребують від вузлів виконання певного обчислювального завдання або криптографічного протоколу. У відповідь на кожний виклик вузол формує спеціальний токен-відповідь (challenge token), який слугує доказом виконання роботи, і цей токен одразу спалюється (знищується) або анулюється. У результаті вузли постійно несуть обчислювальні або фінансові витрати просто для підтримання свого статусу в мережі. Якщо ж вузол не відповість на виклик вчасно, його активність може бути обмежена, що робить атаки типу Сивілли безглуздими — утримування великої кількості фальшивих вузлів вимагатиме пропорційно великих витрат.

Використання RTCT-механізму потенційно дозволяє підвищити рівень безпеки без кардинальної зміни основного протоколу консенсусу. Цей підхід можна розглядати як додатковий «шар» захисту, який комбінується з PoW чи PoS. Наприклад, його доцільно застосувати в PoS-мережах для запобігання ситуації «nothing at stake» та для примушування валідаторів до постійної активності. В PoW-мережах RTCT може ускладнити використання орендованого хешрейту, оскільки потребує тривалого підтримання ресурсів [11], [12].

Постановка проблеми. Аналіз існуючих підходів показав, що базові механізми консенсусу хоча й забезпечують високий рівень безпеки, все ж мають сценарії, у яких мотивований зловмисник може спробувати обійти їх обмеження. Основні проблемні моменти включають:

- Атака Сивілли. У відкритій P2P-мережі створення нових вузлів, як правило, мало що коштує. Це дозволяє одному атакувальнику породити велику кількість псевдовузлів і, наприклад, спробувати заповнити ними мережу, порушуючи маршрутизацію або голосування. Без механізму перевірки «активності» вузлів мережа не відрізняє чесного учасника від вузла-Сивілли [4], [10].
- Оренда ресурсів для атаки. З розвитком хмарних обчислень та спеціалізованих сервісів з'явилася можливість тимчасово отримати значні обчислювальні потужності без придбання у власність. Це ставить під загрозу PoW-мережі з відносно невеликим сумарним хешрейтом — зловмисник може на короткий

період орендувати достатньо потужності для переважання над чесними майнерами. Витрати на таку короткотривалу атаку можуть окупитися, якщо вдасться здійснити подвійне витрачання чи інші шахрайські дії [7].

- «Сплячі» валідатори в PoS. В алгоритмах PoS валідатори отримують винагороди пропорційно їх долі (stake). Однак, якщо валідатор пасивний або рідко обирається для створення блоків, він все одно зберігає свою частку і може потенційно активуватися для атаки (наприклад, спробувати форк після тривалого періоду). Механізм слешингу карає за конкретні зловживання (подвійний підпис тощо), але не змушує валідаторів постійно витрачати ресурси.
- Атаки на рівні мережевого протоколу. Окрім консенсусу, атакувальник може намагатися порушити роботу мережі через флуд-атаки, такі як спам-транзакції або атаки затемнення (ізоляція вузла), які не завжди прямо запобігаються консенсусними механізмами. Економічні витрати на такі атаки також мають значення: наприклад, після впровадження механізму спалення комісії (EIP-1559 в Ethereum) атака спам-транзакціями стала дорожчою, бо базова комісія спалюється і не повертається атакувальнику [2], [9].

Таким чином, постає проблема розробки універсального засобу, що підвищував би поріг витрат для різних типів атак. Цей засіб має:

- Примусово створювати для кожного вузла постійні експлуатаційні витрати (operational costs) навіть за відсутності явної корисної діяльності, щоб виключити «халявних» учасників.
- Робити неможливим масове утримання фальшивих вузлів без відповідних витрат на кожен з них.
- Вводити елемент непередбачуваності в процес перевірки, щоб атакувальник не міг оптимізувати використання ресурсів під графік викликів.
- Легко перевірятися всіма учасниками та бути справедливим (вимоги рівні для всіх вузлів).
- Мінімально впливати на пропускну здатність і затримки основного процесу обробки транзакцій і блоків.

Механізм RTCT покликаний вирішити цю проблему. Його запровадження має на меті ліквідувати «лази» для тих типів атак, які залишаються можливими при традиційних підходах, шляхом додаткового економічного навантаження на потенційного порушника.

Аналіз останніх досліджень і публікацій. Проблемі забезпечення стійкості розподілених систем до економічно мотивованих атак присвячено значну кількість досліджень. У класичній роботі С. Накамото, де запропоновано Bitcoin [1], вперше реалізовано поєднання криптографічного протоколу з економічним стимулом (нагорода за майнінг) як захист від подвійного витрачання та захоплення мережі. Надалі з’явилися альтернативні механізми консенсусу. Зокрема, алгоритм Proof-of-Stake (PoS), активно досліджений В. Бутеріним та ін. у контексті проєкту Ethereum Casper, вводить поняття економічної відповідальності валідаторів: для генерації блоку вони мають внести заставу в криптовалюті, яка буде втрачена у разі нечесних дій. Це забезпечує аналогічний стримуючий ефект, що й PoW, але без прямих енергетичних витрат. Ще одним напрямом є Proof-of-Burn (PoB) — алгоритм, запропонований Айном Стюартом, за яким учасники спалюють власні монети, щоб отримати право майнінгу пропорційно спаленому обсягу. PoB, по суті, моделює витрати як «віртуальне» споживання ресурсів: знищені токени виступають аналогом витраченої електроенергії, піднімаючи бар’єр для атаки.

Окрім консенсусних алгоритмів, досліджуються й інші методи підвищення стійкості мереж. Одним з таких методів є введення періодичних обчислювальних

випробувань для учасників мережі. Наприклад, в роботі Ixian DLT запропоновано механізм Proof-of-Computational-Work (PoCW), за якого кожний вузол періодично виконує ресурсномістке обчислення — пошук хеш-значення із заданою кількістю нулів, та відправляє результат у мережу як частину спеціального «presence»-повідомлення [5]. Таким чином підтверджується активність вузла і накладається додаткове навантаження на потенційного зловмисника, що прагне утримувати багато вузлів. Аналогічні ідеї використання часткового PoW для захисту від спаму і атак Сивілли були раніше реалізовані в системах електронної пошти (наприклад, Hashcash Адама Бека) та мережевих протоколах: кожний запит або дія повинні супроводжуватись обчисленням певного хеш-пазлу, аби ускладнити масові автоматизовані зловживання [6].

Крім класичних способів протидії атакам Сивілли — у бездротових ad-hoc-мережах поширений підхід самореєстрації (self-registration). Узагальнений огляд таких схем подано у [3]. Ідея полягає в поєднанні криптографічного ключа з вимірюваними параметрами каналу (RSSI, часові відмітки), що ускладнює маскування одного фізичного вузла під багатьох логічних. Водночас метод критично залежить від безпосередньої фізичної присутності пристрою й малоефективний у повністю віртуальних P2P-топологіях, які розглядаються в нашій роботі.

Варто зазначити, що значна частина сучасних робіт сфокусована на пошуку балансу між безпекою та ефективністю. Надмірні обчислювальні вимоги можуть знизити продуктивність мережі та відлякати добросовісних учасників. Наприклад, періодичні виклики не повинні бути занадто частими чи важкими, щоб не перевантажити вузли. Деякі дослідження пропонують адаптивне налаштування складності завдань залежно від стану мережі, схоже до регулювання складності в PoW-консенсусі. Інший аспект — генерація справді випадкових викликів. Для цього пропонуються механізми на основі децентралізованих випадкових маяків (random beacons) та функцій з затримкою, що перевіряється (VDF), які гарантують непередбачуваність і достовірність випадкових інтервалів часу [8].

Отже, останні публікації вказують на те, що поєднання криптографічних викликів і економічних стимулів є перспективним шляхом посилення стійкості блокчейн-мереж. Запропонований у цій статті підхід RTCT продовжує цей напрям, розвиваючи ідеї PoW, PoS, PoB та періодичних випробувань, та інтегрує їх у цілісну схему. Нижче детально описано постановку проблеми та сутність механізму RTCT.

Метою статті є формалізація механізму випадкових часових челендж-токенів та аналіз його впливу на криптоекономічну стійкість децентралізованої мережі. Для досягнення мети вирішуються такі завдання: проаналізувати останні дослідження і рішення в сфері криптоекономічної безпеки; описати алгоритм роботи RTCT і необхідні криптографічні примітиви; побудувати математичну модель витрат зловмисника на проведення атаки в мережі з RTCT; дослідити залежність вартості атаки від параметрів RTCT (складності завдання, частоти викликів тощо) і порівняти з базовими моделями; сформулювати висновки щодо ефективності запропонованого підходу та окреслити напрямки подальших досліджень.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Механізм RTCT складається з трьох основних елементів: виклик, токен та спалення. На рис. 1 схематично зображено послідовність дій у рамках RTCT. Виклики генеруються протоколом випадковим чином у різні моменти часу та адресуються вузлам

мережі. Виклик може бути глобальним для всіх вузлів одночасно або індивідуальним для конкретного вузла, вибраного випадково або циклічно. Після отримання виклику вузол повинен вирішити поставлену обчислювальну задачу, результатом чого є випуск челендж-токена — криптографічного доказу виконання задачі.

Рис. 1. Схема механізму RTCT: випадковий виклик → формування токена → спалення токена. Вузол мережі отримує від протоколу випадковий виклик, обчислює відповідь у вигляді челендж-токена та публікує транзакцію спалення, що анулює даний токен

Челендж-токен може мати різну реалізацію залежно від природи виклику. У найпростішому випадку токен — це просто рішення обчислювальної головоломки, такої як пошук хешу з певними властивостями. У більш складних випадках це може бути підпис деяким секретним ключем, який вимагає наявності у вузла певного ресурсу або інформації. Важливо, що всі інші вузли мають змогу перевірити цей токен: тобто існує ефективний алгоритм перевірки, що даний вузол дійсно розв’язав поставлену задачу.

Останній етап — спалення токена. Це означає, що отриманий токен не приносить безпосередньої вигоди тому, хто його згенерував, і не може бути повторно використаний. Якщо токен — це криптовалюта або внутрішня монета мережі, спалення реалізується відправленням її на спеціальну «null-адресу», з якої неможливо витратити кошти. У разі, коли токен є нефінансовим доказом (наприклад, хеш-підпис), «спалення» може полягати в одноразовій реєстрації цього доказу в блокчейні, після чого він не дає подальших прав чи переваг. Головна ідея — вузол витрачає ресурси на отримання токена, але не може повернути ці ресурси або отримати еквівалентну нагороду: токен одразу знецінюється.

Механізм викликів є випадковим у двох аспектах: час і адресація. Для генерації часу викликів використовується надійне джерело випадковості. Це може бути, наприклад, комбінація блочного хешу та криптографічної функції затримки (VDF), щоб результат став відомим лише через заданий інтервал часу, забезпечуючи непередбачуваність моменту виклику [8]. Адресація виклику може бути всієї мережі, що спричиняє масову перевірку, або випадкового підмножини вузлів для розподілу навантаження. Обидва підходи мають переваги: глобальний виклик гарантує однакові умови для всіх, тоді як часткові випадкові виклики унеможливлюють концентрацію атакуючих ресурсів лише на окремих моментах — ніколи не відомо, хто буде викликаний наступним.

Після спалення токена мережа фіксує виконання виклику даним вузлом. Якщо вузол не надав валідного токена у відповідь на виклик протягом відведеного часу, це розглядається як збій або потенційно зловмисна поведінка. Протокол може вжити заходів: знизити репутацію вузла, тимчасово відсторонити його від участі в консенсусі чи навіть стягнути штраф — наприклад, утримати заставу, якщо така передбачена, аналогічно до слешингу в PoS.

Таким чином, RTCT створює постійний «фоновий» процес в мережі, де кожен вузол періодично підтверджує свою надійність через розв’язання випадкових задач. Це схоже на постійне проходження вузлом капчі або тесту на працездатність. Зловмисник, що планує атаку, мусить не лише підготувати основний план (отримати 51% хешрейту чи купити монети), але й підтримувати всі свої вузли активними під час підготовки та здійснення атаки, відповідаючи на RTCT-виклики. В іншому разі його вузли будуть виключені або скомпрометують себе відсутністю відповідей.

МАТЕМАТИЧНА МОДЕЛЬ ВИТРАТ НА АТАКУ МЕРЕЖІ З RTCT

Для того, щоб кількісно оцінити ефект від застосування RTCT, розглянемо модель витрат атакувальника. Припустимо, що без RTCT атакувальник має певні ресурси, які дозволяють йому здійснити атаку із вірогідністю успіху P_0 (можливо близькою до 1, якщо ресурсів достатньо). Введення RTCT змінює ситуацію наступним чином: з'являється додаткове навантаження, яке атакувальник повинен витримувати протягом усього часу атаки.

Позначимо такі параметри моделі:

- m — параметр складності обчислювального завдання у виклику RTCT. Інтуїтивно, m може відповідати, наприклад, кількості провідних нулів у хеші (як у PoW), тобто задача вимагає виконати в середньому 2^m хеш-операцій, або іншому показнику, що експоненційно впливає на складність.
- λ — частота викликів RTCT для кожного вузла (середня кількість викликів за одиницю часу). Для простоти вважатимемо, що кожен вузол отримує виклики з інтенсивністю λ (наприклад, один виклик на годину в середньому, тобто $\lambda = 1/3600\ \text{с}^{-1}$).
- C_{work} — вартість (у USD) виконання однієї одиниці обчислення, нормована на базову складність (наприклад, вартість одного хешу чи іншої елементарної операції). Ця величина залежить від ефективності обладнання і ціни електроенергії.
- N_{attack} — кількість вузлів, якими оперує атакувальник (наприклад, запущених вузлів-Сивілл чи майнерів/валідаторів під його контролем). Для успішної атаки типу Сивілли N_{attack} має становити значну частку від загальної кількості вузлів мережі, а для атаки 51% — достатню частку від загальної потужності/ставки.

Тепер розглянемо сумарні витрати атакувальника за час T реалізації атаки. Витрати складаються з двох складових:

1. Базові витрати на атаку без RTCT — позначимо їх як C_{base}. Це може включати витрати на оренду обладнання, закупівлю монет, виплату хабарів тощо залежно від типу атаки. Ця величина є великою, але в моделі без RTCT ми припускаємо, що атакувальник погоджується її понести заради очікуваного прибутку від атаки.
2. Додаткові витрати RTCT — це витрати на обчислення токенів за кожний виклик для всіх своїх вузлів протягом часу T. Якщо за час T кожен вузол отримує в середньому λT викликів, то кожен вузол має виконати λT задач. Кожна задача складності m має вартість:

$$C_{\text{work}} \cdot 2^m \tag{1}$$

Відповідно, один вузол витрачає за час T:

$$C_{\text{node}}(T) = \lambda T \cdot C_{\text{work}} \cdot 2^m \tag{2}$$

Для N_{attack} вузлів витрати атакувальника становлять:

$$C_{\text{RTCT}}(T) = N_{\text{attack}} \cdot \lambda T \cdot C_{\text{work}} \cdot 2^m \tag{3}$$

Таким чином, загальна вартість атаки з урахуванням RTCT:

$$C_{\text{total}}(T) = C_{\text{base}} + C_{\text{RTCT}}(T) \tag{4}$$

Важливим спостереженням є те, що C_{RTCT} зростає лінійно з N_{attack} експоненційно з параметром складності m. Базова вартість C_{base} може бути різною для різних атак, але додаткова компонента має універсальний характер.

Розглянемо відносне зростання вартості атаки завдяки RTCT. Нехай атакувальник для успіху повинен контролювати частку p від мережі або володіти p долею ставки чи хешрейту. Це може означати, що частка від загальної кількості вузлів:

$$N_{\text{attack}} = pN_{\text{total}} \quad (5)$$

Можна вважати, що C_{base} росте як функція від лінійно або більш швидко. Додаткові витрати RTCT однозначно ростуть пропорційно p:

$$C_{\text{RTCT}} \approx pN_{\text{total}} \cdot \lambda T \cdot C_{\text{work}} \cdot 2^m \quad (6)$$

Навіть якщо атакувальник намагатиметься зекономити, зменшуючи число активних вузлів, він не зможе опуститися нижче порогу, потрібного для успіху атаки — інакше атака не має сенсу. Отже, p є фіксованим — близьким до 0.5 для атаки-51% або визначене умовами успіху атаки Сивілли.

Значення λT залежить від тривалості атаки. Якщо атака короткочасна, T мале, але навіть за короткий період відбудеться певна кількість викликів. У граничному випадку миттєвої атаки (що практично неможливо, оскільки блокчейн-атаки вимагають часу на майнінг блоків чи підтвердження транзакцій) RTCT не встигне накласти істотних витрат. Проте реалістично атака триває хоча б декілька годин чи днів, протягом яких RTCT нарощує витрати для нападника.

В якості ілюстрації, припустимо $\lambda = 1$ виклик на годину для кожного вузла, C_{work} дуже мале (на порядки менше цента за хеш), але m досить велике (наприклад, $m = 40$, що еквівалентно ~1 трильйону хешів на виклик). Тоді один вузол витрачає на годину:

$$C_{\text{node}} \approx C_{\text{work}} \cdot 2^{40} \quad (7)$$

Якщо $C_{\text{work}} = 10^{-16}$ USD, то $C_{\text{node}} \sim 2^{40} \cdot 10^{-16} USD \sim 1.1 \cdot 10^6$ USD (близько одного мільйона доларів на годину). Це гіпотетичний приклад з дуже високою складністю. Але він показує принцип: експоненційне зростання витрат зі збільшенням m.

На рис. 2 наведено графік залежності логарифму вартості атаки C_{RTCT} від параметра складності m при фіксованих інших параметрах. Графік є близьким до лінійного, оскільки на логарифмічній шкалі вплив 2^m проявляється як лінійна функція ($\log_{10}(C_{\text{RTCT}}) \sim m \log_{10} 2$), якщо інші множники постійні.

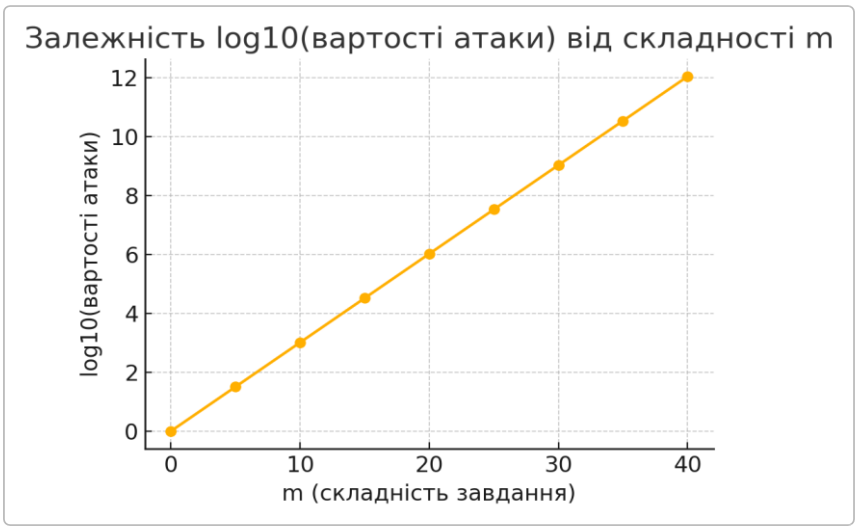

Рис. 2. Залежність логарифму вартості атаки від параметра складності m. Показано експоненційне зростання витрат: кожне збільшення m на 1 приводить до подвоєння необхідних ресурсів атакувальника (лінійний ріст по осі Y, оскільки відкладено log10(вартості))

ISSN 2663 - 4023

Формально це можна записати як:

$$\log_{10} C_{\text{RTCT}} = \log_{10}(N_{\text{attack}} \cdot \lambda T \cdot C_{\text{work}}) + m \cdot \log_{10} 2 \quad (8)$$

Таким чином, нахил прямої на рис. 2 визначається $\log_{10} 2 \approx 0.3010$ (якщо C_{work} та інші фактори постійні). На графіку видно, що навіть помірне збільшення складності (наприклад, з $m = 10$ до $m = 20$) піднімає вартість на кілька порядків. З точки зору вибору параметрів, мережа може підлаштовувати m так, щоб досягти бажаного балансу: зробити атаку практично нездійсненною, але і не створити зайвого тягаря для чесних вузлів. Зазвичай орієнтиром є економічна доцільність: витрати на атаку мають перевищувати потенційну вигоду настільки, щоб навіть найбільш ресурсні зловмисники були відлякані. Якщо, наприклад, максимальна раціональна сума, яку хтось може витратити на атаку, оцінюється у X доларів, то параметри RTCT повинні бути такими, щоб $C_{\text{RTCT}}(T)$ для часу атаки T перевищувало X.

Отримані оцінки є наближеними, реальні атаки можуть бути багатофакторними, а зловмисники можуть шукати оптимізації, наприклад, розподіляючи виклики між своїми вузлами, вимикаючи частину вузлів з меншими ролями тощо. Однак будь-які відхилення від виконання RTCT-викликів ведуть до того, що деякі вузли нападника не підтвердять свою активність і будуть виключені з процесів мережі, знижуючи шанс на успіх атаки. Тому для спрощення можна вважати, що зловмисник все одно змушений обслуговувати всі свої вузли.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Вплив RTCT на учасників мережі та параметри системи. Впровадження RTCT впливає не лише на зловмисників, а й на чесних учасників. Очевидно, що всі вузли тепер мають додаткове навантаження, тож необхідно проаналізувати, чи не стане це надмірним тягарем для них та чи не призведе до централізації. Наприклад, лише багаті учасники зможуть дозволити собі працювати вузлом.

Розглянемо чесний вузол. Він теж отримує виклики з інтенсивністю λ і повинен витрачати ресурси, еквівалентні $C_{\text{node}}(T)$ за час T. На відміну від зловмисника, чесний вузол отримує користь від нормальної роботи в мережі – наприклад, майнери отримують нагороди, валідатори — комісії, вузли — доступ до послуг мережі тощо. Отже, для чесного учасника RTCT — це додаткова «операційна витрата», яка зменшує чистий прибуток від участі. Якщо ця витрата занадто велика, частина потенційних учасників можуть відмовитись від запуску вузла, що негативно вплине на децентралізацію. Тому вибір λ і m повинен враховувати середні можливості чесних вузлів. У сучасних блокчейн-мережах вузли часто запускаються на комерційних серверах чи хмарних платформах; можна закласти, що середній вузол має певну кількість вільних обчислювальних ресурсів (CPU/GPU) поза основним завданням. RTCT має споживати цей «запас» без критичного впливу на основну діяльність. Наприклад, якщо вузол завантажений консенсусом на 50%, RTCT може займати ще 10-20% чи менше, залишаючи резерв.

Управління складністю може здійснюватися динамічно. Аналогічно до того, як у Bitcoin кожні 2016 блоків регулюється складність PoW задля підтримання стабільного часу блоку, в RTCT можна коригувати m залежно від статистики виконання викликів. Якщо більшість вузлів відповідають занадто швидко і легко, можна підняти складність; якщо ж фіксується багато пропущених відповідей навіть від явно чесних вузлів, слід складність знизити або зменшити частоту λ. Щодо частоти λ, її вибір визначає компроміс між оперативністю контролю та накладними витратами. Висока λ (часті виклики) забезпечує швидке виявлення неактивних чи зловмисних вузлів, але значно збільшує

сумарні обчислення в мережі. Занадто низька λ зменшує ефективність RTCT — зловмисник може встигнути здійснити коротку атаку між викликами. Практично, λ можна прив'язати до часу генерації блоків. Наприклад, можна реалізувати політику: кожен валідатор (або майнер) у проміжках між створенням блоків має отримати один випадковий виклик. У такому випадку λ пропорційна швидкості блокчейну.

Інший важливий аспект — сумісність RTCT з протоколом консенсусу. У PoW-мережі RTCT виглядає як додатковий незалежний PoW-процес. Це може дещо знизити ефективність основного PoW (майнер частину потужності витрачає на RTCT-виклики, а не на майнінг блоків). Проте якщо RTCT-головоломка подібна до PoW-задачі (наприклад, пошук низького хешу), майнер може використовувати те саме обладнання, перемикаючись між завданнями. У PoS-мережах RTCT додає вимогу виконувати роботу навіть тоді, коли валідатор не обирається для блоку, що є принциповою відмінністю від чистого PoS. Це усуває проблему «ледачого стейкера» і робить протокол ближчим до PoW за витратами, але без масового дублювання роботи — кожен вузол робить свою невеличку роботу, а не змагається за один блок.

Приклад розрахунку та сценарій атаки. Розглянемо гіпотетичну мережу і порівняємо два сценарії: з RTCT і без нього. Нехай мережа — PoS-блокчейн з капіталізацією 1 млрд USD, у стейкінгу знаходиться 60% монет (600 млн USD). Зловмисник розглядає атаку, скуповуючи частку монет. Без RTCT він повинен купити >50% стейка, тобто понад 300 млн USD, щоб мати шанс контролювати мережу (це C_{base}). Це вже велика сума, але якщо мережа молода, інвестор з такими коштами теоретично може спробувати (у деяких атаках на менш вартісні блокчейни витрачали десятки мільйонів). Тепер припустимо, що мережа впровадила RTCT з параметрами: один виклик кожні 10 хвилин ($\lambda = 6/\text{год}$) з завданням, еквівалентним $m = 30$ (приблизно 10^9 хеш-операцій) і $C_{\text{work}} = 2 \cdot 10^{-12}$ USD за операцію (що відповідає витратам електроенергії на ASIC). Тоді один вузол витрачає на годину: $C_{\text{node, hour}} = 6 \cdot 10^9 \cdot 2 \cdot 10^{-12} = 12 \cdot 10^{-3} = 0.012$ USD. За добу це всього ~0.288 USD. Ця величина майже не помітна для чесного валідатора (кілька центів витрат на добу). Але атакувальник, який скупив 51% монет, має, припустимо, 1000 вузлів (для надійності розподілив свою частку на багато адрес). За добу його витрати на RTCT ~ 288 USD. За місяць ~ 8640 USD. Це все ще малі гроші порівняно з $C_{\text{base}} = 300$ млн USD. Однак картина зміниться, якщо атакувальник спробує використати інший вектор — атака Сивілли з малим стейком. Наприклад, замість того, щоб купувати 300 млн, він намагається створити 10,000 вузлів з мінімальним стейком (скажімо, по 1000 USD кожний, разом 10 млн USD) і сподівається захопити більшість вузлів у голосуваннях на рівні мережі (де вага може бути 1 вузол = 1 голос в деяких протоколах). Без RTCT ця атака можлива, якщо протокол погано спроєктований і покладається на кількість вузлів. З RTCT кожен з 10,000 вузлів потребує обслуговування викликів. При $C_{\text{node, day}} \approx 0.288$ USD, загальні витрати за добу становлять 10,000 * 0.288 = 2880 USD. За рік ~1.05 млн USD тільки на обчислення. Це суттєво змінює економіку атаки: підтримка ботнету з 10,000 вузлів стає доволі дорогою. І головне – ці витрати є постійними, їх не можна разово інвестувати і потім вимкнути обладнання: атакувальник повинен продовжувати платити протягом усього часу, поки він хоче тримати свої вузли активними в мережі.

Цей приклад демонструє, що RTCT особливо ефективний проти широкомасштабних атак Сивілли і довготривалих спроб контролю мережі. Для короткої атаки (наприклад, викуп 51% і негайна спроба зміни правил) додаткові витрати хоча й присутні, але можуть здатись не такими значними порівняно з величезною базовою інвестицією. Втім, у поєднанні зі слешингом ризик для атакувальника стає подвійним: він витрачає сотні мільйонів на монети, плюс несе операційні витрати, і може втратити все це в разі невдачі атаки та покарання.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У даній статті запропоновано та досліджено механізм випадкових часових челендж-токенів (RTCT) як інструмент підвищення криптоекономічної стійкості децентралізованих мереж. Проведений аналіз показав таке:

- Існуючі консенсусні алгоритми PoW, PoS, PoB забезпечують базовий рівень криптоекономічної безпеки, проте зловмисники можуть намагатися використати певні прогалини (ідентичності Сивілли, оренда потужностей, пасивні валідатори) для атак.
- Механізм RTCT доповнює протокол мережі постійними випадковими перевірками працездатності кожного вузла, що створює додаткові безпекові гарантії. Кожен вузол змушений регулярно вирішувати обчислювальну задачу і несе часові, енергетичні або фінансові витрати на її вирішення, не отримуючи прямого прибутку.
- Математична модель витрат на атаку з урахуванням RTCT продемонструвала експоненційне зростання сукупних витрат для атакувальника при збільшенні складності викликів. Це підтверджується графічно: залежність log10(вартості) від параметра складності близька до лінійної, що відповідає геометричному росту вартості в абсолютному вираженні. Збільшення складності m дозволяє довільно підвищити ціну атаки, обмежуючись лише тим, щоб чесні вузли ще могли виконувати поставлені задачі.
- Впровадження RTCT робить атаки типу Сивілли економічно невигідними навіть при невеликій складності, оскільки витрати зловмисника ростуть прямо пропорційно до кількості підконтрольних вузлів. Навіть якщо кожен виклик коштує частку цента, масова мережа фальшивих вузлів обходитиметься дуже дорого протягом тривалого часу.
- Проаналізовано вплив параметрів RTCT на чесних учасників. Показано, що при розумному налаштуванні навантаження на вузол буде прийнятним і не відлякає доброчесних операторів. Ключовим є динамічне налаштування параметрів: мережа може автоматично підлаштовувати складність задач, виходячи з цільового рівня безпеки та спостережуваної пропускної здатності вузлів.

Таким чином, RTCT є перспективним доповненням до існуючих механізмів забезпечення безпеки блокчейн та інших децентралізованих систем. Він не замінює консенсус, але створює додатковий рівень захисту, який ускладнює реалізацію як технічних (обчислювальних), так і економічних атак.

Перспективи подальших досліджень включає наступні напрямки:

– Імітаційне моделювання: розробка програмного симулятора мережі з RTCT для оцінки впливу на час підтвердження транзакцій, пропускну здатність та поведінку учасників в умовах різних атак.
– Оптимізація розкладу викликів: дослідити, як найкраще генерувати випадкові виклики – можливо, нерівномірними інтервалами, сплесками тощо – щоб максимізувати дезорієнтацію зловмисника, зберігаючи ефективність.
– Комбінація зі слешингом і стимулюванням: розглянути варіанти, коли за пропущені RTCT-виклики учасник не тільки тимчасово відключається, а й втрачає заставу або репутацію, а навпаки за стабільне виконання отримує невеликі нагороди. Це може збалансувати витрати чесних вузлів.

– Апаратні аспекти: проаналізувати, які типи обчислень для RTCT найлегше інтегрувати у вже існуюче майнінгове або валідаторське обладнання. Наприклад, використання ASIC для подвійної роботи: і для основного консенсусу, і для RTCT.
– Інтеграція в реальні протоколи: спробувати впровадити прототип RTCT у тестову мережу популярного блокчейну чи в середовище симуляції, таку як Ethereum тестнет зі зміненими правилами, та виміряти реальний вплив на метрики безпеки.

Спираючись на результати цього дослідження, можна зробити висновок, що поєднання криптографічних викликів з економічними стимулами та штрафами є дієвим шляхом підвищення стійкості децентралізованих систем. Механізм RTCT розвиває цю ідею, надаючи простий у реалізації, але потужний інструмент для ускладнення роботи потенційним атакувальникам. Подальша робота в цьому напрямку дозволить уточнити параметри й адаптувати механізм до різних архітектур децентралізованих мереж, зробивши екосистему Web 3.0 ще безпечнішою.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system.* https://bitcoin.org/bitcoin.pdf
2. Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. (2015). Eclipse attacks on Bitcoin's peer-to-peer network. *In Proceedings of the 24th USENIX Security Symposium*, 129–144.
3. Vasudeva, A., & Sood, M. (2018). Survey on Sybil attack defense mechanisms in wireless ad hoc networks. *Journal of Network and Computer Applications, 120,* 78–118. https://doi.org/10.1016/j.jnca.2018.07.006
4. Douceur, J. R. (2002). The Sybil attack. In IPTPS 2002, 251–260.
5. Finney, H. (2004). *RPOW – Reusable proofs of work* [White paper].
6. Back, A. (2002). *Hashcash – A denial-of-service counter-measure* [Technical report].
7. Laurie, B., & Clayton, R. (2004). Proof of work proves not to work; version 0.2. *In Workshop on Economics of Digital Security (WEEDS'04).*
8. Boneh, D., Bünz, B., & Fisch, B. (2018). A survey of verifiable delay functions (VDFs). *IACR Cryptology ePrint Archive, 2018/601.*
9. Wood, G. (2014). *Ethereum yellow paper: A formal specification of the Ethereum virtual machine* (Version 9f6b).
10. Cosman, B., Fisch, B., & Shen, C. (2020). *Escrow-based cryptoeconomic incentives for Sybil-resistant consensus.* arXiv preprint arXiv:2006.12943.
11. Sompolinsky, Y., & Zohar, A. (2015). Secure high-rate transaction processing in Bitcoin. *In Financial Cryptography and Data Security, 8975,* 507–527).
12. King, S., & Nadal, S. (2012). *PPCoin: Peer-to-peer crypto-currency with proof-of-stake* [White paper].

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (28), 2025
ISSN 2663 - 4023

Ivan Parkhomenko
Candidate of Technical Sciences, Associate Professor, Head of CSIP Department
Faculty of Information Technology
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
ORCID ID: 0000-0001-6889-9284
ivan.parkhomenko@knu.ua

Roman Ohiievych
Postgraduate at CSIP Department
Faculty of Information Technology
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
ORCID ID: 0009-0003-7948-1125
ohiievychr@fit.knu.ua

CRYPTO-ECONOMIC RESILIENCE OF A DECENTRALIZED NETWORK USING RANDOM TIME CHALLENGE TOKENS (RTCT)

Abstract. The paper presents a novel approach to enhancing the crypto-economic resilience of decentralized networks by employing a mechanism of Random Time Challenge Tokens (RTCT). Contemporary consensus mechanisms (Proof-of-Work, Proof-of-Stake, Proof-of-Burn) and their ability to deter 51% attacks and Sybil attacks are analyzed. It is shown that classical approaches secure the network by imposing substantial economic barriers to attackers – for example, miners are forced to invest in hardware and electricity, raising the cost of a 51% attack to a prohibitively high level. However, vulnerabilities remain: attackers can rent computational resources or exploit under-active validators. The proposed RTCT mechanism involves generating random cryptographic challenges at unpredictable time intervals, which network nodes must answer with a cryptographically verifiable token, followed by burning of that token. This process creates unpredictable load and continuous costs for network participants, significantly increasing the economic cost of attacks. The RTCT process is mathematically formalized and the dependence of attack cost on the challenge complexity parameter m is evaluated. The results confirm that as complexity m increases, an attack requires exponentially higher expenditures, making the network more resilient. The advantages and potential drawbacks of the RTCT approach are discussed, as well as possible directions for further research, including optimizing challenge frequency and integrating RTCT with existing consensus protocols.

Keywords: crypto-economic resilience; decentralized network; challenge token; Proof-of-Work; Proof-of-Stake; Proof-of-Burn; 51% attack; Sybil attack; spam attack.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system.* https://bitcoin.org/bitcoin.pdf
2. Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. (2015). Eclipse attacks on Bitcoin's peer-to-peer network. *In Proceedings of the 24th USENIX Security Symposium*, 129–144.
3. Vasudeva, A., & Sood, M. (2018). Survey on Sybil attack defense mechanisms in wireless ad hoc networks. *Journal of Network and Computer Applications, 120,* 78–118. https://doi.org/10.1016/j.jnca.2018.07.006
4. Douceur, J. R. (2002). The Sybil attack. In IPTPS 2002, 251–260.
5. Finney, H. (2004). *RPOW – Reusable proofs of work* [White paper].
6. Back, A. (2002). *Hashcash – A denial-of-service counter-measure* [Technical report].
7. Laurie, B., & Clayton, R. (2004). Proof of work proves not to work; version 0.2. *In Workshop on Economics of Digital Security (WEEDS'04).*
8. Boneh, D., Bünz, B., & Fisch, B. (2018). A survey of verifiable delay functions (VDFs). *IACR Cryptology ePrint Archive, 2018/601.*
9. Wood, G. (2014). *Ethereum yellow paper: A formal specification of the Ethereum virtual machine* (Version 9f6b).
10. Cosman, B., Fisch, B., & Shen, C. (2020). *Escrow-based cryptoeconomic incentives for Sybil-resistant consensus.* arXiv preprint arXiv:2006.12943.

11. Sompolinsky, Y., & Zohar, A. (2015). Secure high-rate transaction processing in Bitcoin. *In Financial Cryptography and Data Security*, *8975*, 507–527).
12. King, S., & Nadal, S. (2012). *PPCoin: Peer-to-peer crypto-currency with proof-of-stak*e [White paper].