НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
Навчально-науковий фізико-технічний інститут
Кафедра математичних методів захисту інформації

«На правах рукопису»
УДК (впишіть правильний УДК!)

«До захисту допущено»
Завідувач кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2026 р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Математичні методи криптографічного захисту інформації»

зі спеціальності: 113 Прикладна математика
на тему: **«Прискорене моделювання однієї з стратегій атаки розгалуження на блокчейн, який використовує протокол консенсусу Proof-of-Stake»**

Виконав:
студент II курсу, групи ФІ-23
Кіяшко Ігор Володимирович _____

Керівник:
старший викладач, кандидат фіз.-мат наук
Кузнєцов Ігор Миколайович _____

Рецензент:
провідний науковий співробітник відділу математичних методів теорії надійності складних систем Інституту кібернетики ім. В.М.Глушкова НАН України, доктор фіз.-мат. наук
Коба Олена Вікторівна _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ — 2026

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
Навчально-науковий фізико-технічний інститут
Кафедра математичних методів захисту інформації

Рівень вищої освіти — перший (бакалаврський)
Спеціальність — 113 Прикладна математика,
ОПП «Математичні методи криптографічного захисту інформації»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2026 р.

ЗАВДАННЯ
на дипломну роботу

Студент: Кіяшко Ігор Володимирович

1. Тема роботи: «*Прискорене моделювання однієї з стратегій атаки розгалуження на блокчейн, який використовує протокол консенсусу Proof-of-Stake*», науковий керівник дисертації: старший викладач, кандидат фіз.-мат наук Кузнєцов Ігор Миколайович,

затверджені наказом по університету №__ від «__» _____ 2026 р.

2. Термін подання студентом роботи: «__» _____ 2026 р.

3. Об'єкт дослідження: процес формування розгалужень у блокчейн-системах з протоколом консенсусу Proof-of-Stake під впливом стратегічної поведінки зловмисника

4. Предмет дослідження: математичні моделі, алгоритми та програмні засоби прискореного моделювання атаки розгалуження в блокчейн-системах з протоколом консенсусу Proof-of-Stake

5. Перелік завдань:

– аналіз існуючих атак на блокчейн-системи та особливостей атак розгалуження

– дослідження принципів функціонування протоколу консенсусу Proof-of-Stake

– побудова математичної моделі атаки розгалуження

– розробка та реалізація алгоритму прискореного моделювання

– реалізація алгоритму моделювання методом Монте–Карло

– верифікація реалізованих алгоритмів шляхом порівняння з результатами, наведеними в науковій літературі

– проведення експериментальних досліджень для різних параметрів моделі

– аналіз ефективності прискореного моделювання порівняно з методом Монте–Карло

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: презентація доповіді, графіки залежності ймовірності успішного розгалуження від параметрів моделі, блок-схеми алгоритмів, результати чисельних експериментів у вигляді таблиць та діаграм.

7. Орієнтовний перелік публікацій: планується доповідь на всеукраїнській науковій конференції

8. Дата видачі завдання: 10 вересня 2025 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	01-15 вересня 2025 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	Вересень-жовтень 2025 р.	Виконано
3	...	...	Виконано

Студент _____ Кіяшко І.В.

Керівник _____ Кузнєцов І.М.

РЕФЕРАТ

Кваліфікаційна робота містить: 75 стор., 11 рисунки, 8 таблиць, 32 джерел.

Мета роботи це реалізація алгоритму прискореного моделювання атаки розгалуження в блокчейн-системах з протоколом консенсусу Proof-of-Stake. Також метої є аналіз імовірнісних характеристик розгалужень при різних значеннях параметрів моделі.

Об'єктом дослідження є процес формування розгалужень у блокчейнах із протоколом консенсусу Proof-of-Stake під впливом раціональної стратегії поведінки зловмисника.

Предметом дослідження є математичне та програмне моделювання атаки розгалуження з використанням алгоритму прискореного моделювання.

У роботі доведено теоретично і практично, ефективність прискореного моделювання для аналізу рідкісних подій. Проведено реалізації, верифікацію та експерименти, які показали всю ефективність.

КЛЮЧОВІ СЛОВА: БЛОКЧЕЙН, PROOF-OF-STAKE, АТАКА РОЗГАЛУЖЕННЯ, SPLITTING ATTACK, МОНТЕ-КАРЛО, FAST SIMULATION, МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, ІМОВІРНІСНИЙ АНАЛІЗ.

ABSTRACT

The qualification work contains: 75 pages, 11 figures, 8 tables, 32 sources.

The purpose of the work is to implement an algorithm for accelerated modeling of a branching attack in blockchain systems with the Proof-of-Stake consensus protocol. The purpose is also to analyze the probabilistic characteristics of branches for different values of the model parameters.

The object of the study is the process of forming branches in blockchains with the Proof-of-Stake consensus protocol under the influence of the attacker's rational behavior strategy.

The subject of the study is mathematical and software modeling of a branching attack using the accelerated modeling algorithm.

The work proves theoretically and practically the effectiveness of accelerated modeling for the analysis of rare events. Implementations, verification and experiments were carried out, which showed all the effectiveness.

ЗМІСТ

ВСТУП

Два десятиліття тому блокчейн-технології вважалися тільки вульзкоспеціалізованою інновацією у сфері криптографії. Але зараз вони відіграють все більше і більше важливу роль для безпеки інформації, допомогають управляти цифровими активами, фінансовими системами та навіть розповсюджені у питаннях цифрового врядування. В 2008 році, з появою біткойну, концепція розподіленого реєстру значно еволюціонувала [1]. Сучасні мережі все частіше і частіше переходять від енергновитраного протоколу консенсусу Proof-of-Work до набагато більш ефективного протоколу Proof-of-Stake. Незважаючи на те, що це підвищує маштабованість, це також відкриває нові вектори атак. На даний момент існує багато нових протокіл. Сам їх огляд наведено у роботі [2]. З неї можна дізнатися, що кількість різноманітних атак на структуру блокчейн є доволі різноманітною та значною. Сама перша відома атака була атакою подвійної витрати (Double Spend Attack), вона описана в історичній роботі S.Nakamoto [1]. Мета цієї атаки це використати одну і ту же монету для різних розрахунків. А одна з найнебезпечніших атак на блокчейн це атака розгалуження (Splitting Attack) [3], про яку ця дипломна робота буде більш детально розглядати.

Більше класичних атак має ціль підробити транзакції, або змінити якісь дані. Але ціль у атаки розгалуження це створити ситуацію, в якій мережа не має точної впевіненості в тому, яка гілка головна. Через це створюється невизначеність у мережі. При цьому лінійна гілка блокчейну розщеплюється на дві гілки, з яких обидні гілки можуть вважатися валідними. Ця атака дуже сильно ускладнює верифікацію транзакцій і також дозволяє зловмиснику впливати на стабільність системи. Ця атака не спричиняє прямої втрати активів, але через неї встрачається довіра до мережі і може знизити вартіть відповідної криптовалюти.

Суть цієї атаки полягає у тому, щоб зловмисник, який має частку

стейку, одночасно підтримував дві однакові гілки. І якщо система обере довшу або випадкову з цих двох, то зловмисник може навмисно створити "двозначність"у структурі блокчейну. Така проблема особливо критична для фінансових додатків, бо ставить під загрозу незворотність транзакцій.

Інколи буває таке, що розгалуження з'явилось і без атаки зловмисника. Це нормально, бо, наприклад, синхронізація не може бути ідеальною в цьому випадку. З моменту ствоерння та адсилання блоку до моменту, коли цей блок отримають всі учасники мережі може багато що трапитись. Може бути таке, що блок матиме дві гілки, через які і з'явиться це розгалуження. Якщо виникли дві гілки, то за правилами майнінгу кожен наступний блок приєднується до довшої гілки, якщо ж вони мають однакову довжину, то з рівною ймовірністю до будь-якої. Зауважимо, що більш коротка гілка в історії блокчейну не зберігається (тобто щезають всі транзакції у цій гілці, а також монети, зароблені слотлідерами). Коректною вважається більш довга гілка.

Для однієї з атак по типу подвійної витрати у багатьох випадках вдається обчислити її ймовірність за допомогою аналітичних формул. А ось для атаки розгалуження прямих формул не існує. Відомі лише асимптотичні оцінки [4], основним недоліком яких є неможливість оцінити їх точність (точність можна розцінити як прийнятну лише для достатньо великої довжини блокчейну).

Найбільше популярним в інженерній практиці став метод Монте Карло [5 – 8] моделювання імовірнісних характеристик безпосередньо на моделі системи чи мережі, що досліджується. Така популярність методу Монте-Карло визначається трьома його властивостями: a) можливістю враховувати практично всі характерні властивості моделі; b) відносною простотою реалізації; c) високою швидкістю побудови незміщеної оцінки в одній реалізації алгоритму. Розвиток комп'ютерних технологій та обчислювальних потужностей також сприяв поширенню цього методу. Хоча метод Монте-Карло є наближеним методом, він дає змогу будувати асимптотичні довірчі інтервали, які практично завжди накривають

невідому ймовірнісну характеристику, що оцінюється.

Проте метод Монте-Карло має єдиний, але дуже суттєвий недолік: час моделювання імовірнісної характеристики із заданими достовірністю та відносною похибкою є обернено пропорційним до цієї ймовірності. Це означає практичну неможливість використання методу Монте-Карло (занадто велика тривалість моделювання для досягнення потрібної точності оцінки). У цьому випадку можна піти трьома шляхами: або при обчисленнях збільшити відносну похибку, або обчислення проводити на багатопроцесорному комплексі, або застосувати один із методів зменшення дисперсії оцінки (variance reduction techniques).

Перші два шляхи не потребують зміни алгоритму моделювання. Набагато більш цікавим є третій шлях, в основі якого лежить ідея зменшення дисперсії оцінки при збереженні її незміщеності (це так зване прискорене моделювання – Fast Simulation). Бурхливий розвиток різноманітних підходів до зменшення дисперсії оцінки спостерігається протягом декількох останніх десятиліть. Найбільшу популярність отримав метод суттєвої вибірки (importance sampling) [9 – 12]. У той же час цей метод потрібно використовувати з певною обережністю, оскільки при формальному використанні дисперсія оцінки може не тільки не зменшитись, але й стати нескінченною. Серед інших методів прискореного моделювання варто відзначити аналітико-статистичний метод, розроблений українською школою математичної теорії надійності під керівництвом академіка НАН України І.М.Коваленка [6, 13], метод розшарованої вибірки [14, 15], метод багаторівневого розщеплення [16] та низку інших [5, 6, 10, 17, 18].

Актуальність дослідження. Дослідження являється актуальним, бо блокчейн-технології дуже активно використоруються в економічній, адміністративній та правовій сферах. Через високу потребність в використанні, я прогресує потреба, щоб були інструменти аналізу. Вони повинні добре тестувати стікість до атак і прогнозувати атаки навіть в критичних ситуаціях. У цієї дипломної роботі розглядається раціональна

стратегія поведінки зловмисника під час атаки розгалуження на протокол консенсусу PoS, коли він при додаванні кожного блока вирішує, чи варто саме в цей момент розпочинати розгалуження. Досліджується ймовірність того, що довжина розгалуження досягне певної величини. Також тут показується практичне застосування методу прискореного моделювання, який поєднує в собі як елементи методу Монте-Карло, так і рекурентні формули (аналітична частина) для оцінки ймовірності досягнення певної глибини розгалуження. А коректність запропонованого підходу тут перевіряється через порівняння на числових прикладах оцінок, отриманих прискореним моделюванням та стандартним методом Монте-Карло. Із збільшенням глибини розгалуження метод Монте-Карло стає непридатним для практичного використання, оскільки швидко зростає час моделювання для досягнення потрібної точності. У залежності від оцінюваної імовірності, метод прискореного моделювання дає виграш у часі, що може складати декілька порядків.

Об'єктом дослідження Об'єктом дослідження в дипломній роботі є процес формування розгалужень у блокчейнах із протоколом консенсусу Proof-of-Stake під впливом раціональної стратегії поведінки зловмисника.

Предметом дослідження Предметом дослідження є математичне та програмне моделювання атаки розгалуження з використанням алгоритму прискореного моделювання.

textbfМетою дослідження Мета роботи полягає в реалізації та верифікації цього алгоритму, а також в аналізі імовірнісних характеристик розгалужень при різних вхідних параметрах. Для досягнення поставленої мети передбачено вирішення наступних завдань:

1) провести аналіз існуючих типів атак на блокчейн та виділення особливостей атак розгалуження;

2) розробити формалізовану модель поведінки зловмисника у межах заданої стратегії;

3) розробити та реалізувати алгоритм прискореного моделювання, а також алгоритм моделювання методом Монте-Карло;

4) верифікувати коректність запропонованих алгоритмів шляхом порівняння на числових прикладах оцінок, отриманих прискореним моделюванням та стандартним методом Монте-Карло;

5) на конкретних прикладах проілюструвати перевагу прискореного моделювання порівняно з методом Монте-Карло.

У роботі використовуються методи теорії ймовірностей, рекурентні співвідношення, метод Монте-Карло та програмне моделювання мовою Python. За основу досліджень було взято результати робіт [19, 20], у яких запропоновано математичну модель атаки розгалуження та підхід до її прискореного моделювання.

Практичне значення результатів полягає у тому, що завдяки суттєвому зменшенню часу моделювання з'являється можливість з високою точністю (наприклад, з відносною похибкою 1%) моделювати малоймовірні події (наприклад, обчислювати ймовірність того, що розгалуження досягне певної глибини при великих значеннях). Результати роботи можна використати у наукових дослідженнях і також в інженерії, для проектування блокчейн-протоколів.

Наукова новизна отриманих результатів полягає у тому, що запропоновано новий підхід до моделювання атаки розгалуження при використанні протоколу консенсусу Proof-of-Stake. Новий алгоритм моделювання грунтується на використанні ідей методу суттєвої вибірки при обчисленні ймовірності того, що розгалуження досягне певної глибини.

1 АНАЛІЗ АТАКИ РОЗГАЛУЖЕННЯ У БЛОКЧЕЙНАХ НА ОСНОВІ ПРОТОКОЛУ КОНСЕНСУСУ PROOF-OF-STAKE

Цей розділи буде розповідати основу роботи Proof-of-Stake, як працюють транзакції та їх підтвердження. Також дуже детально будуть розбиратися атаки на блокчейн, їх принцип роботи та ціль використання. Особливої уваги тут буде виділятися атака розгалуження (splitting attack), чому саме вона буде використовуватись для цієї роботи. Після всього в розділі буде інформації про метод Монте-Карло та причини чому він дуже неефтивний, якщо порівнювати з методом Fast Simulation.

1.1 Загальні принципи функціонування протоколу Proof-of-Stake

Сама ідея консенсусу у блокчейн-системах в наш час це базова основа. Вона визначає спосіб, у який учасники децентралізованої мережі досягають згоди щодо порядку та змісту транзакцій. У класичних реалізаціях, таких як Bitcoin, використовується протокол Proof-of-Work (PoW). В ньому право створити наступний блок отримує той учасник, який першим виконає складне обчислення. Тобто той, хто знайде певне значення хешу, отримає це право. Цей процес енерговитратний і, хоча добре захищений від атак, має серйозні обмеження у масштабованості та екологічності. Як відповідь на ці недоліки, було запропоновано альтернативний підхід – Proof-of-Stake (PoS) [1].

Зазвичай у протоколах PoS учасник, який має право створювати блоки (надалі будемо називати його валідатором) обирається не на основі обчислювальних потужностей, а з урахуванням кількості криптовалюти, яку він заморожує (стейку) у системі. Якщо пояснювати простіше, то PoS надає можливість впливу на консенсус пропорційно до володіння

активом, що функціонує в межах блокчейну. Такий підхід дозволяє значно знизити енергоспоживання системи, зменшити інтервали між блоками та підвищити загальну пропускну здатність мережі. Але в цей варіант дій створює нові типи уразливості, які виникають через особливості розподілу контролю в системі між її учасниками.

У PoS-системах саме значення часу поділяється на епохи та таймслоти. Епоха - це відрізок часу, який складається з фіксованої кількості таймслотів, а у кожному таймслоті призначається слотлідер – стейкхолдер, який має право запропонувати новий блок. Призначення відбувається на основі ймовірнісного механізму, в якому шанси бути обраним прямо пропорційні до частки валідатора в загальному пулі стейку. Якщо, наприклад, певний учасник володіє 10 відсотоками стейку, то ймовірність бути обраним слотлідером у будь-якому окремому таймслоті дорівнює 0.1. Розподіл слотлідерів зазвичай фіксується заздалегідь для всієї епохи. Це дозволяє учасникам підготуватися до своєї ролі, а також передбачити (у рамках допустимого рівня прозорості) поведінку інших гравців.

Щоб перевірити право на то, хто може створити блок в PoS, в більшості випадків використовується механізм VRF (Verifiable Random Function). Саме він дозволяє учаснику у приватному режимі перевірити, чи є він слотлідером у заданому таймслоті. При цьому функція VRF генерує значення, яке потім можна криптографічно перевірити іншим учасникам. Завдяки цьому забезпечується достовірність, але без розкриття надлишкової інформації. Сам факт обрання не потребує централізованої перевірки – це повністю децентралізований і верифікований процес [21].

Отже спотлідер створює блок, після чого включає до нього списки транзакцій і головне це підпис, який доводить його право бути лідером у цьому слоті Далі блок передається в мережу, де інші учасники перевіряють його правильність за кількома критеріями: відповідність правил формування блоку, правильність посилання на попередній блок

(геш-послідовність), правильність транзакцій, відсутність подвійних витрат, коректність криптографічного підпису тощо.

Всі інші вузли відразу додають блок до своєї копії блокчейну, як тільки сам блок визнається валідним. У разі, коли кілька блоків претендують на один і той самий таймслот або попередній блок, мережа використовує правило довшої гілки. Це правило означає, що валідною вважається гілка блоків, яка має найбільшу сумарну вагу (у спрощеній реалізації – просто довжину). У випадку, коли довжина двох гілок однакова, рішення приймається випадковим або псевдовипадковим способом. І саме ця ситуація – наявність двох однакових за довжиною гілок – є точкою входу для потенційної атаки розгалуження.

У такому контексті важливо зрозуміти, що мережа не має способу відрізнити чесного слотлідера від зловмисника в певних випадках. Наприклад, він може легально створити два різних блоки в одному слоті, кожен з яких посилається на інший попередній блок. Якщо обидва блоки мають правильні підписи й формально відповідають протоколу, то вони визнаються валідними. Надалі можуть виникнути випадки, коли частина мережі може обрати один блок, а інша – другий. Це утворить утворюючи розгалуження (або по-іншому форк). Якщо така ситуація виникає рідко, то це не є проблемою. Але зловмисник може свідомо утримувати або відтворювати систематичну проблему і через це виникне постійна невизначеність.

Також потрібно сказати, що в PoS є особливість пов'язана з тим, що учасники вимушені зберігати цілісність системи. Їх стейки можуть зникнути у випадку серйозних порушень консенсусу. Тобто учасник не хоче втратити частину або весь свій депозит, тому і буде зберігати цілісність. Проте це стримує лише ті дії, які можна чітко довести як порушення. Дії, що формально є допустимими, але мають руйнівний ефект (наприклад, створення кількох валідних гілок) часто залишаються у сірій зоні, що відкриває потенціал для атак стратегічного типу.

Система використовує такі параметри як глибина підтвердження

(кількість блоків, після яких транзакція вважається незмінною), частота появи слотлідерів, тривалість епох. Ці парметри прямо впливають на стійкість системи до розгалужень. Наприклад, якщо підтвердження вимагає лише кілька блоків, але зловмисник здатен утримувати розгалуження довжиною у десяток слотів, це створює ризик скасування вже підтверджених транзакцій. У такому випадку втрачається гарантія незворотності транзакцій – ключова перевага блокчейну порівняно з централізованими базами даних [22].

Можна сказати, що Proof-of-Stake – це складна й ефективна система, яка дозволяє досягати консенсусу з меншими витратами ресурсів. Але треба розуміти, що вона потребує особливої уваги до моделей безпеки.

1.2 Консенсус та механізми підтвердження транзакцій

Основа для будь-якої блокчейн-системи це механізм досягнення згоди між децентралізованими вузлами. Вони діють незалежно і не довіряють одне одному. Цей протокол консенсусу визначає, як саме мережа приходить до єдиного бачення історії транзакцій. Консенсус не лише управляє дією вузлів, а й забезпечує незмінність даних, їх послідовність і довіру до результатів обчислень у середовищі з потенційно зловмисними учасниками.

Історично першим практичним алгоритмом консенсусу був Proof-of-Work (PoW). Але з часом все більше популярності набуває Proof-of-Stake (PoS). Саме він знижує енергоспоживання і дозволяє досягати консенсусу без інтенсивної роботи обчислювальних пристроїв. У межах PoS ключову роль відіграє поняття стейку – кількості монет або токенів, які учасник заморожує у системі як заставу. Розмір стейку визначає ймовірність того, що саме цей учасник стане слотлідером і отримає право створити наступний блок у визначеному таймслоті. Таким чином, вплив на консенсус розподіляється відповідно до економічної

участі, що стимулює добросовісну поведінку.

Якщо розглянути процес самого консенсусу у PoS, то можна зрозуміти, що він має частку тимчасову структуру. Саме в ньому час поділяється на послідовні таймслоти, які об'єднуються в епохи. У кожному таймслоті зайджи генерується хоча б один блок. який потім додається до лянцюга. Відповідно розподіл прав на створення блоку у слоті робиться, використовуючи всевдовипадкові алгоритми. Основна з них це VRF (Verifiable Random Function) і вона гарантує і випадковість і перевірюваність без розкриття приватної інформації. Завдяки цьому механізму, кожен вузел може незалежно перевірити, чи має він право, щоб створити блок в конкретному слоті.

Коли вузол створює блок, він додає до нього криптографічний підпис, який підтверджує його ідентичність як обраного слотлідера. Блок містить посилання на попередній блок (через хеш), а також перелік транзакцій, які слід підтвердити. Інші вузли, отримавши цей блок, перевіряють його дійсність: коректність підпису, цілісність транзакцій, зв'язність із попереднім блоком. Якщо перевірка пройдена успішно, блок включається до локальної копії ланцюга [23].

Важливо підкреслити, що у розподіленому середовищі блоки можуть надходити до різних вузлів із затримкою. Унаслідок чого в один і той самий таймслот можуть з'явитися кілька альтернативних блоків. Така ситуація не є помилкою, а є особливістю децентралізованих мереж. Для її розв'язання застосовується правило, згідно з яким правильною вважається найдовша гілка. Іншими сломави, та що містить найбільше блоків, починаючи від спільного предка і є правильною гілкою. Це правило дозволяє усім вузлам зрештою синхронізувати свої ланцюги. Це працює, навіть в тому випадку, якщо на проміжному етапі вони мають відмінну локальну історію блоків.

Проблема виникає тоді, коли дві гілки мають однакову довжину. У такому разі стандартний підхід полягає у випадковому виборі однієї з них. Це тимчасове розгалуження зазвичай зникає, щойно в одному зі слотів буде створено блок лише на одній з гілок, і вона автоматично стане

довшою. Однак до цього моменту мережа перебуває в стані невизначеності: транзакції в обох гілках вважаються умовно підтвердженими, а їх остаточна доля залежить від подальшого розвитку подій.

Мехізм, який підтверджує транзакції у таких умоваї, являється прогресивним. Іншими словами, чи більше блоків буде додано поверх перної транзакції, тим більше є її стікість до скасування. Це породжуеє поняття глибини підтвердження

Твердження 1.1. *Транзакція вважається остаточною, якщо її глибина перевищує деякий поріг.*

Вибір цього порогу залежить від багатьох факторів, зокрема від імовірності виникнення розгалужень, швидкості підтвердження блоків і типу застосування. У фінансових системах, наприклад, зазвичай чекають підтвердження декількома блоками перед тим, як вважати транзакцію остаточною [24].

Конфліктні ситуації виникають тоді, коли рівновага між продуктивністю і безпекою зміщується. З одного боку, користувачі хочуть мінімізувати час очікування, а з іншого – збільшення ризику скасування транзакцій неприйнятне. Це особливо актуально в умовах атак, коли зловмисник навмисно створює ситуацію з однаковими гілками. У PoS системах така атака є реалістичною, оскільки обраний слотлідер може легітимно згенерувати кілька блоків, не порушуючи очевидних протоколів. Якщо його частка в мережі досить велика, він зможе впливати на розвиток гілок у такий спосіб, щоб утримувати тимчасову невизначеність якнайдовше.

У такій ситуації механізми підтвердження втрачають передбачуваність. Користувач, який отримав підтверджень, не може бути впевненим у незворотності транзакції, оскільки паралельна гілка, з якої зловмисник тимчасово відмовився, може раптово стати головною. Цей феномен лежить в основі атаки розгалуження і безпосередньо залежить від механізмів консенсусу та параметрів мережі.

У протоколах PoS передбачено додаткові механізми захисту, такі як штрафи (slashing). Саме вони застосовуються до вузлів, що порушують правила. Прикладом, можна взяти ситуацію, коли підписують кілька блоків у тому самому слоті. Проте якщо зловмисник діє в межах дозволених дій або використовує недосконалості протоколу, виявити і покарати його неможливо. Це створює простір для стратегічної поведінки, що формально не є порушенням, але ставить під загрозу цілісність ланцюга [25].

Отже можна сказати, що сама концепція підтвердження транзакцій у PoS це не якийсь легкий технічний механізм. Це ціла структура, яка дарує баланс між швидкістю, надійністю та безпекою. Системи мають або використовувати глибші пороги підтвердження, або оцінювати ризики динамічно. Це все залежить від історії мережі, частоти появи конфліктів, поведінки валідаторів. Такі речі формують основу для математичного моделювання процесів розгалуження. Можна навіть виділити випадки, коли атака є результатом раціональної поведінки окремого учасника мережі, незважаючи на нечесність самої поведінки.

Системи мають або запроваджувати глибші пороги підтвердження, або оцінювати ризики динамічно – залежно від історії мережі, частоти появи конфліктів, поведінки валідаторів. Усе це формує підґрунтя для математичного моделювання процесів розгалуження – зокрема у випадках, коли атака є результатом раціональної, хоча й нечесної, поведінки окремого учасника мережі.

1.3 Теоретичні основи та класифікація атак на блокчейн

Хочеться відзначити, що безпека в блокчейн-систем дуже важлива та відрізняється від безпеки звичайних систем. Вона поєднує криптографічну стійкість, стратегії розподілу довіти йо ймовірнісні моделі поведінки учасників. А у звичайних системах захист залежить від централізованої політики та адміністрування. В нашому випадку система

повинна вміти передбачати наявність зловмисних вузлів. А в випадку їх знаходження негайно нейтралізувати за рахунок економічної, математичної та криптографічної структури мережі. Це означає, що блокчейн потребує глибокої теоретичної підготовки, щоб вміти захиститися від зловмисників. Потрібно добре вміти аналізувати моделі ймовірності, стратегічну поведінку та формалні гарантії узгодженості [26].

Загалом атаки на блокчейн використовуються для певних цілей. В деяких випадках мета атаки - отримати фінансову вигоду, в інших - прагення дестабілізувати мережу. Також можливі сценарії, коли зловмисник хоче провести атаку, щоб маніпулювати самими даними. Незалежно від мети або цілі атаки, використовують різні способи атаки. Але в більшості випадки їх об'єднує те, що ціль атаки здійснити її не через грубе порушення, а через використання його специфіки.

Також дуже важливо розуміти рівні моделей, щоб знайти як вони працюють та розуміти як відбувається втручання. Наприклад, є атаки на мережевому рівні. Вони включають в себе затримку або блокування повідомлень між вузлами. Також є атаки на рівні консенсусу, де зловмисник намагається впливати на сам вибір гілки або порядок блоків. Ще один варіант це атаки економічного характеру. У рамках цієї класифікації особливе місце займають атаки, пов'язані з розгалуженнями блоків.

Одна з найвідоміших атак має назву 51%. В ній зловмисник намагається побудувати альтернативний ланцюг блоків, враховуючи те, що він контролює більшість обчислювальної (у PoW) або стейкової (у PoS) потужності. Якщо така гілка перевищує довжину основної, то вона визнається мережею як валідна. Завдяки цій схемі зловмисник може скасовувати вже підтверджені транзакції. Загалом, завдяки цьому, він може реалізувати атаку подвійної витрати. Загалом це критична атака і для неї навіть існує багато аналітичних оцінок. Але незважаючи на ці проблеми, її практично неможливо реалізувати, бо вона потребує неймовірної кількості ресурсів [27].

Також дуже цікавими є атаки на фіналізацію транзакцій. Такі типи атак не майже не потребують якихось ресурсів. Наприклад, існує атака нічого не коштує (nothing-at-stake attack). Її можна використати, коли у PoS мережі вузли можуть підтримувати одразу кілька гілок. Якщо з часов одна з гілок виявиться основною, то вузол отримає винагороду. Цей сценарій не залежить від того, якою була його поведінка. Це створює парадок - чесна поведінка не є обов'язково раціональною. Сама мережа стає вразливою до розгалужень навіть без централізованого контролю зловмисника.

Більш витонченими є атаки на фіналізацію транзакцій. Ця атака не потребує абсолютної більшості ресурсів. Наприклад, атака нічого не коштує (nothing-at-stake attack) виникає тоді, коли у PoS мережі вузли мають стимул підтримувати одразу кілька гілок, адже це не потребує додаткових витрат. У разі, якщо з часом одна з цих гілок виявиться основною, вузол отримає винагороду, незалежно від того, якою була його поведінка. Це створює парадоксальну ситуацію, у якій чесна поведінка не є обов'язково раціональною. Сама мережа стає вразливою до розгалужень навіть без централізованого контролю зловмисника.

Іншим прикладом є лонг-рейндж атаки (long-range attacks). Вони реалізуються завдяки використанню старих приватних ключів, які були активними на початку існування мережі. У разі втрати контролю над поточним стейком зловмисник може будувати окремий ланцюг, що розпочинається з давно минулої епохи. Він буде намагатися переконати нові вузли в його валідності. Такі атаки особливо небезпечні для вузлів, які щойно підключилися до мережі й не мають повної історії блоків. Вони можуть довірити свою синхронізацію фальшивій гілці, що була створена заднім числом і через це потрапити під повний контроль зловмисника [28].

Ще слід розповісти про атаку розгалуження, яка поєднує особливості кількох підходів, які ми розглядали раніше. Ідея цієї атаки поляє у тому, щоб підтримувати незавершеність підтвердження. Наприклад, банківські розрахунки або смарт-контракти з часовою

прив'язкою. Це особливо небезпечно для інфраструктури, яка залежить від незмінності записів. Атака працює так, що зловмисник намагається підтримувати існування відразу двох гілок. Така схема буде працювати, навіть якщо він має відносно невеликий стейк. Якщо гілок стає декілька, то мережа не зможе відразу визначити, яка з них стане основною. Через це частина учасників мережі може використовувати різні версій блокчейну. Якщо такий стан буде зберігатися протягом певного часу, то це значно збільшить довжину розгалуження. І ця проблема буде загрожувати скасуванням вже підтверджених транзакцій.

Треба зазначити, що такий в цій атаці не потрібно прямо порушувати протокол. Все, що робить зловмисник, формально є нормальним відносно поставлених правил. Незважаючи на це, в сукупності вони створюють руйнівний вплив на стабільність мережі. Це робить її особливо небезпечною. Традиційні механізми виявлення порушень, по типу штрафів або блокуваннь тощо, не завжди можна застосувати. Аналіз такої атаки потребує моделювання поведінкових стратегій. Вони не зводяться до простих бінарних дій (чесно/нечесно), а являються наслідком оптимізаційної задачі на стороні зловмисника [29].

Класифікація атак також охоплює стохастичні характеристики. Це стосується ймовірності успішного завершення атаки, середньої довжини гілки, частоту появи сприятливих умов для початку розгалуження тощо. На цьому моменті методи аналізу стають математичними. Основна мета це побудова моделей, які дозволяють з високою точністю оцінити ризики.

Можна сказати, що теорія атак на блокчейн охоплює широкий спектр стратегій. Вона потребує гарного розуміння, як все працює та які атаки і. Саме тому ефективне проектування блокчейн-системи неможливе без глибокого розуміння потенційних векторів атак і математичних методів оцінки їхньої ефективності.

1.4 Механізм, наслідки, економічна мотивація атаки розгалуження

Необхідно обов'язково розібрати атаку розгалуження (splitting attack). Вона належить до того типу атак, який реалізується не через пряме порушення правил протоколу.Загалом атака просто використовує доступні механізми в своїх цілях. У контексті блокчейн-систем, що використовують протокол Proof-of-Stake, вона набуває особливої актуальності. Це все через те, що використовувати атаку, можна навіть, якщо зловмистик не має більшу частину стейку, як в атаці 51%. Навіть без цього, зловмисник може впливати на довжину та стабільність лацюга блоків. Головна мета атаки, щоб так само створити невизначеність в мережі. Це дозвонить ускладнити досягнення консенсусу й поставить під сумнів незворотність підтверджень [30].

В основі самої атаки полягає проста задумка. Уявімо сценарій, де зловмисник може організувати ситуацію, у якій мережа матиме дві однакові за довжиною гілки блоків. І якщо трапиться так, що він буде тримати її протягом часу, то всі ті транзакції з гілок, будуть вважатися непідтвердженими, поки все не вирішиться. У реальності це означає, що будь-яка транзакція, може бути скасована. Це трапиться незалежно від кількості підтверджень і тільки якщо виявиться, що гілка програла конкуренцію. І якщо людина не має впевненість, що вже навіть підтвердженну транзакцію не скасують, то вона не буде використовувати та довіряти такій системі.

Механізм реалізації атаки починається з декількох кроків. Спочатку зловмисник, який має певну частку стейку, чекає на момент, коли він двічі поспіль отримує можливість створити блок (наприклад, за схемою 0–1–0, де 0 – зловмисник, 1 – чесний учасник). Далі, з такою конфігурацією він спочатку створює блок, який продовжує основну гілку. Потім зловмисник дозволяє чесному вузлу додати свій блок. Після цього

створює ще раз блок, який відгалужується від попереднього. Внаслідок цього утворюється розгалуження довжиною два блоки. Надалі зловмисник використовує кожен свій слот для підтримки обох гілок. Він буде чергувати створення блоків у них. Якщо у них однакова довжина, то буде дублвати свої дії. Таким чином зберігається ситуація, коли жодна з гілок не перемагає. Це буде тривати до тих пір, поки один із чесних вузлів не внесе зміни до рівноваги, або зловмисник не втратить контроль [31].

Наслідки такого впливу є комплексними. По-перше, транзакції не можуть завершитись. Поки одну з гілок не визнають дійсною, тобто вона виграє і стане головною, то користувачі не операції не будуть важатися завершеними. Це підриває основну перевагу блокчейну - незмінність записів, яка лежить в основі його цінності для цифрових контрактів, розрахункових систем і захищених реєстрів. По-друге, мережа втрачає ефективність. Це пояснюється тим, що частина згенерованих у процесі розгалуження блоків виявляються марними і не потрапляють до фінального ланцюга. Це означає, що ресурси обчислення, пропускна здатність і час були витрачені даремно. Крім того, тривале існування кількох гілок, які буквально конкрують, що стати головними, зменшує довіру користувачів до мережі та ускладнює використання блокчейну для практичних задач.

Мотивація зловмисника застосувати таку атаку може бути варіативною. Вона не обмежується тим, щоб отримати гроші прямо. Однією з причин проведення такої атаки може бути бажання понизити репутацію або стабільність роботи блокчейн-системи. Зловмисник може створити атаку, щоб знизити довіру. Після цього він купує активи за зниженими цінами. Така схема дозволяє отримати значний прибуток без фізичного викрадення коштів або порушення очевидних правил. Інший варіант – атака як інструмент боротьби з конкурентами. Тобто коли одна мережа намагається дискредитувати іншу. Наприклад, щоб привабити її користувачів чи інвесторів до власної екосистеми [32].

Особливо цікавою є ситуація, коли атака розгалуження маскується

під нормальну поведінку мережі. У мережах, де немає єдиного центру управління, виникають затримки під час поширення інформації між різними вузлами. Особливо це трапляється у великих системах, якими користуються тисячи учасників по всьому світу. Через такі затримки часто виникають ситуації, коли існують кілька варіантів історії блоків. Це відбувається тому, що різні частини мережі можуть мати різну інформацію про останні зміни в своїй історії. Така ситуація є швидше звичайним явищем, ніж рідкістю. Тому важко зрозуміти чи це був зловмисник чи ні. А сам зловмисник може скористатися цим, щоб уникнути покарання. З точки зору протоколу, всі дії виглядають допустимими. Блоки мають правильну структуру, підписані коректними ключами, відповідають тимчасовим рамкам і правилам вибору попередника. Проте на стратегічному рівні вся ця поведінка спрямована на затримку остаточного формування консенсусу.

Тому під час розробки протоколів консенсусу важливо враховувати можливість появи та тривалого існування розгалужень. Одне з можливих рішень – запровадження механізмів економічного стимулювання, які заохочують чесну поведінку або штрафують за підтримку кількох гілок одночасно. Інше – побудова аналітичної моделі, яка дозволяє оцінити ймовірність виникнення розгалуження за заданих параметрів. Це дає змогу обрати глибину підтвердження, за якої ризик скасування транзакції є прийнятно низьким [3].

У системах Proof-of-Stake атака розгалуження має певні особливості, якщо будемо повірнювати з Proof-of-Work. У протоколах на основі обчислювальної складності дублювання зусиль коштує дорого. Натомість у PoS створення кількох блоків в одному таймслоті – не така вже й дорога операція, особливо якщо вузол не ризикує втратити депозит. Це призводить до ситуації, коли дешевизна атаки компенсується її стратегічною ефективністю, а низький поріг входу робить її привабливою навіть для учасників із відносно малим стейком.

З математичного погляду, аналіз атаки розгалуження зводиться до

побудови моделі, яка враховує послідовність таймслотів, розподіл лідерства між вузлами, стратегію зловмисника, правила продовження гілок, а також початкові умови розгалуження. Для дослідження таких процесів використовуються математичні моделі, які описують появу блоків та зміну стану розгалуження з часом. Таке моделювання дозволяє оцінити ключову характеристику – імовірність того, що розгалуження досягне певної довжини. Чим меншою є ймовірність виникнення довгого розгалуження, тим вищою вважається стійкість блокчейн-системи до таких атак [19].

Таким чином, атака розгалуження – це не просто теоретичний сценарій, а реальна загроза для практичної реалізації консенсусу у децентралізованих мережах.

1.5 Огляд існуючих моделей та досліджень. Метод Монте-Карло та прискорене моделювання: переваги та недоліки

Із ростом інтересу до протоколів консенсусу на основі Proof-of-Stake зросла й потреба у точному кількісному аналізі їх безпеки. Особливе місце в цьому контексті займає вивчення стратегічних атак, зокрема атак розгалуження, які, на відміну від прямого порушення правил, реалізуються шляхом допустимих, але зловмисно скоординованих дій.

Стандартні підходи до вивчення таких атак включають методи стохастичного моделювання, симуляції на основі генерації послідовностей подій, а також побудову рекурентних співвідношень для обчислення ймовірності успіху атаки. Усі ці підходи мають спільну мету — отримати надійні числові оцінки ймовірності того, що атака досягне певного рівня розвитку, зокрема глибини розгалуження [20].

Метод Монте-Карло

Початкові дослідження цієї теми базувалися на методі Монте-Карло, який передбачає численну генерацію випадкових послідовностей дій вузлів

у мережі та статистичний аналіз отриманих результатів. Його перевагами є простота реалізації, універсальність та гнучкість щодо типу моделі.

Метод Монте-Карло дозволяє будувати незміщені оцінки ймовірностей та отримувати асимптотичні довірчі інтервали для характеристик, що оцінюються.

Стисло розглянемо математичне підґрунтя цього підходу.

Закон великих чисел

Теорема 1.1 (Теорема Хінчина)**.** *Нехай*

$$X_1, X_2, \ldots, X_n$$

— незалежні та однаково розподілені випадкові величини зі скінченним математичним сподіванням

$$\mathbb{E}[X_i] = a.$$

Тоді для будь-якого $\varepsilon > 0$

$$\lim_{n\to\infty} P\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - a\right| > \varepsilon\right) = 0.$$

Тобто

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{P} a.$$

Наслідок 1.1 (Наслідок теореми Колмогорова)**.** *Нехай*

$$X_1, X_2, \ldots, X_n$$

— незалежні та однаково розподілені випадкові величини зі скінченними математичним сподіванням a *та дисперсією* σ^2*.*

Тоді

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{\text{п.н.}} a.$$

Тобто середнє арифметичне збігається до математичного

сподівання майже напевно.

Якщо оцінювана ймовірність p визначається методом Монте-Карло, то випадкові величини

$$X_i = \begin{cases} 1, & \text{якщо подія відбулася,} \\ 0, & \text{інакше.} \end{cases}$$

У цьому випадку

$$\mathbb{E}[X_i] = p,$$

а оцінка ймовірності має вигляд

$$\hat{p} = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

Центральна гранична теорема

Для побудови довірчих інтервалів використовується центральна гранична теорема.

Теорема 1.2 (Центральна гранична теорема)**.** *Нехай*

$$X_1, X_2, \ldots, X_n$$

— незалежні та однаково розподілені випадкові величини зі скінченними математичним сподіванням a та дисперсією σ^2.

Тоді

$$\lim_{n\to\infty} P\left(\frac{\sum_{i=1}^{n} X_i - na}{\sigma\sqrt{n}} < x \right) = \Phi(x),$$

де $\Phi(x)$ — функція Лапласа (стандартна нормальна функція розподілу).

Оцінка необхідної кількості реалізацій

Нехай необхідно забезпечити відносну похибку оцінки

$$\frac{|\hat{p} - p|}{p} \leq \varepsilon$$

з достовірністю

$$\gamma.$$

Тоді кількість реалізацій визначається формулою

$$N = \left\lfloor \frac{u_\gamma^2 (1-p)}{\varepsilon^2 p} \right\rfloor + 1,$$

де u_γ визначається з рівняння

$$\Phi(u_\gamma) = \frac{1+\gamma}{2}.$$

Для малих значень p

$$N \approx \frac{u_\gamma^2}{\varepsilon^2 p}.$$

Наприклад, якщо

$$\gamma = 0.99, \qquad u_\gamma \approx 2.576,$$

та

$$\varepsilon = 0.01,$$

то

$$u_\gamma^2 \approx 6.64.$$

Якщо оцінювана ймовірність є малою, наприклад

$$p = 10^{-8},$$

то необхідна кількість реалізацій стає надзвичайно великою.

Недоліки методу Монте-Карло

Основним недоліком методу Монте-Карло є різке зростання часу моделювання під час оцінювання малих імовірностей. У таких випадках можливі три підходи:

1) збільшення допустимої похибки ε;

2) використання багатопроцесорних обчислювальних систем;

3) застосування методів зменшення дисперсії оцінки.

Перші два підходи не потребують зміни алгоритму моделювання, тоді як третій дозволяє суттєво прискорити процес обчислень.

Прискорене моделювання

Ідея прискореного моделювання полягає у зменшенні дисперсії оцінки без втрати її незміщеності. Такі методи відомі як *variance reduction techniques*.

У цій роботі пропонується поєднати елементи методу Монте-Карло з рекурентними формулами обчислення імовірнісних характеристик розгалуження. Такий підхід дозволяє значно скоротити час моделювання та зберегти необхідну точність оцінок.

Коректність підходу перевіряється шляхом порівняння результатів прискореного моделювання з результатами класичного методу Монте-Карло.

Асимптотичні оцінки

Важливу роль у дослідженні атак розгалуження відіграють асимптотичні оцінки.

Для частки зловмисника, меншої за критичне значення, ймовірність досягнення глибини розгалуження L експоненційно спадає зі зростанням L:

$$P(L) \sim Ce^{-\lambda L},$$

де $C > 0$ та $\lambda > 0$ — сталі параметри моделі.

Такі оцінки дозволяють визначати необхідну глибину підтвердження

транзакцій, проте не дають точних значень імовірностей для конкретних параметрів системи.

Сучасні підходи до аналізу атак розгалуження

Сучасні дослідження безпеки PoS-протоколів спрямовані на створення точних числових моделей, придатних для практичного використання. Одним із найбільш ефективних підходів є побудова рекурентних моделей на основі двовимірного представлення станів системи та таблиць переходів між ними.

Такі методи дозволяють:

– зменшити використання пам'яті;

– скоротити час обчислень;

– отримувати точні оцінки для конкретних параметрів мережі;

– реалізовувати алгоритми у вигляді програмного забезпечення для аналізу безпеки блокчейн-систем.

Висновки

Отже, у сучасній науковій літературі сформувалася багаторівнева система підходів до аналізу атак розгалуження — від класичних статистичних симуляцій до складних рекурентно-аналітичних моделей.

Найбільш перспективним підходом є поєднання методу Монте-Карло з рекурентними обчисленнями та методами зменшення дисперсії. Саме такий підхід реалізується в даній роботі, що дозволяє оцінити його практичну ефективність та провести дослідження для нових наборів параметрів, які не розглядалися у попередніх роботах.

Висновки до розділу 1

Зараз можна підвести підцумок до першого розділу дипломної роботи. В ній було пояснено теоретичну частину блокчейн-систем із консенсусом Proof-of-Stake та як саме вона функціонує. Також було проаналізовано структуру атак на них та детально пояснено атаки розгалуження. Було наведено аргументи, чому вона є однією

найскладніших і водночас малодосліджених загроз для таких систем. Особливо показано, що навіть дотримуючись формальних правил в мережі, можна дестабілізувати її та провести атаку, щоб зменьшити довіру до системи.

Переглянувши наукові публікації і дослідження, можна впевненно сказати, що методи аналізу таких атак активно еволюціонують. І можна зробити висновок, що класичному метод Монте-Карло застарілий, якщо порівнювати з Fast Simulation. Ми показати, наскільки вона краще і ефективніше не лише для оцінити ймовірність успіху атаки для конкретних вхідних параметрів, але і для широкого аналізу. Це дуже корисно та важдиво для створення блокчейн-систем.

Показано необхідність практивно реалізувати цей аналітичний метод, щоб підвищити рівень безпеки для блокчейну. Це допоможе покращити підтвердження транзакцій та зменшити ситуації неоднозначності, через яку може зникнути довіра до цілої системи.

2 МАТЕМАТИЧНА МОДЕЛЬ ТА АЛГОРИТМ ПРИСКОРЕНОГО МОДЕЛЮВАННЯ

В цьому розділі буде теоретично показано як саме зловмисник буде проводити атаку, Як визначаються спотлідери та отримується частка блоку. Та які саме формалами можна використати, щоб проаналізувати стратегії початку розгалуження. Більш детально буде приділятися увага формулам та теоретичної реалізації, щоб показати ефективніть алгоритму прискореного моделювання.

2.1 Формалізація стратегії зловмисника у моделі атаки

Загалом, в протоколі консенсусу Proof-of-Stake валідатори мають право створити нові блоки. Їх кількість залежить від кількості заблокованих (стейканих) токенів. Для аналізу безпеки потрібно приділяти увазі не тільки математичній структурі протоколу, також потірнбо не забувати про поведінку самого зловмисника, якій теж має частину стейку. Атака розгалужження (splitting attack) являється дуже небезпечною атакою, бо суть атаки полягає у тому, щоб утимати дві або більше гілок блокчейнку. В цілому вони утримуються в синхронізованому, але неузгодженому стані.

Для того щоб формалізувати саму стратегію, потрібно нам потрібно розглянути процес створення блоків у мережі. Його можна уявити як послідовність окремих часових інтервалів (слотами). У кожному слоті обирається лідер, що отримує право створити новий блок. Можемо розглянути модель Ouroboros. В протоколі вибір такого лідера відбувається за допомогою криптографічних механізмів, зокрема функції VRF (Verifiable Random Function). На цій основі і будується розклад майбутніх лідерів. Ця логіка може бути використана не тільки

звичайними учасниками, а й зловмисниками, щоб той зміг використати її для побудови власної альтернативної гілки блокчейну. При цьому він не обов'язково створює її одночасно з основною гілкою. Зловмисник може випереджати основний ланцюг або навпаки навмисно затримувати публікацію своїх блоків, що ускладнює виявлення розгалуження іншими учасниками мережі.

Ціль моделі полягає у тому, щоб зловмисник підтримував дві гілки. Першла гілга буде відповідати офіційному ланцюгу, а ось друга буде лише його приватною і тільки він зможе її контролювати. Зловмисник буде чергувати публікацію блоків, щоб мережа не змогла зрозуміти, яка гілка правильна. Проблема в процесі полягаю в тому, що дуже важливо слідкувати за часом публікації самого блоку, не можна використовувати лише зміст.

Розглянемо математичну модель атаки розгалуження у протоколі Proof-of-Stake.

Означення 2.1. Нехай p — ймовірність того, що черговий слот отримує чесний учасник, а q — ймовірність того, що слот отримує зловмисник. Припустимо, що в кожному слоті може бути створений лише один блок.

Нехай злодій контролює частку стейку α, тоді

$$q = \alpha, \qquad p = 1 - \alpha.$$

Означення 2.2. Стан системи на кроці n описується парою

$$(i,j),$$

де i — довжина приватної гілки зловмисника, а j — довжина публічної гілки.

Означення 2.3. У кожному слоті система може перейти до одного з двох нових станів:

$$(i,j) \to (i+1,j),$$

якщо слот отримує зловмисник, та

$$(i,j) \to (i,j+1),$$

якщо слот отримує чесний валідатор.

Позначимо через p ймовірність того, що в певному слоті блокує чесний учасник, і через $q = 1 - p$ - відповідно, ймовірність появи блоку з боку зловмисника.

У класичній моделі атаки вважається, що лише один блок може бути згенерований у слоті. Нехай зловмисник володіє часткою α від загального стейку системи, а решта $1 - \alpha$ належить чесним валідаторам. Тоді

$$q = \alpha, \qquad p = 1 - \alpha.$$

Процес моделюється як двовимірний дискретний процес, у якому стан системи на n-му кроці позначається парою (i,j), де i - довжина приватної гілки зловмисника, а j - довжина публічної гілки.

У кожному слоті можливі два сценарії:

– зловмисник отримує слот і продовжує приватну гілку:

$$(i,j) \to (i+1,j);$$

– чесний учасник отримує слот і розвиває офіційну гілку:

$$(i,j) \to (i,j+1).$$

На рисунку 2.1 схематично зображено структуру переходів між станами у просторі (i,j), де стрілки вказують на можливі переходи за результатами чергового слота.

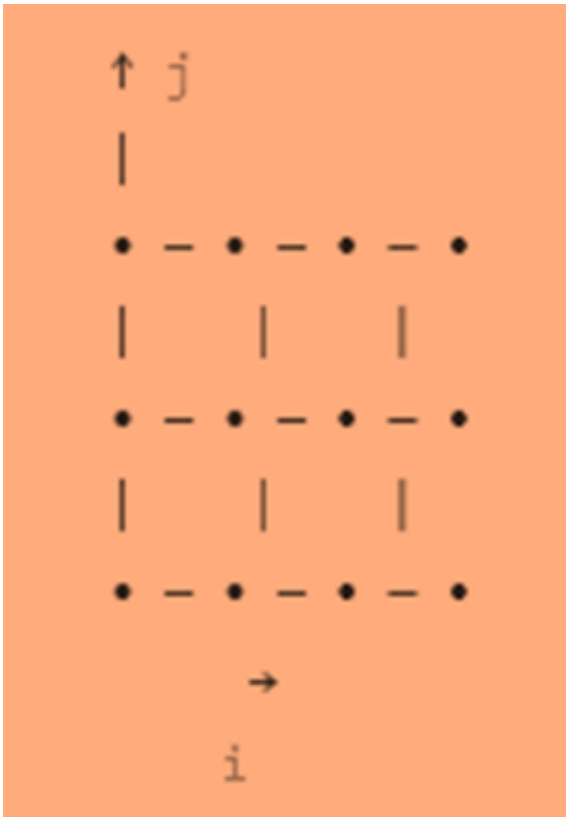

Рисунок 2.1 – Модель простору станів розгалуження

Кожна точка на картинці - це стан. А стрілки показують можливі переходи: вгору - чесний, вправо - зловмисник.

Щоб розгалуження пройшло успішно, зловмиснику потрібно виконати певні умови. Одне з них це довжина його приватної гілки повинна бути не мешна або рівна як довжина офіційної. Ціль якраз і є видати альтернативну гілку так, щоб при публікації мережа прийняла її як валідну.

Означення 2.4. Атака розгалуження досягла глибини L, якщо різниця між довжиною приватної гілки зловмисника та довжиною публічної гілки дорівнює значенню L, тобто

$$i - j = L,$$

де $L \in \mathbb{N}$ — бажана глибина розгалуження.

Параметр L використовується як міра успішності атаки. Досягнення

певної глибини розгалуження надає зловмиснику можливість впливати на процес підтвердження та фіналізації транзакцій.

α впливає на саму ймовірність розгалуження, що воно досягне потрібної глибини L. Але крім на це ще вливає конкретна стратегія поведінки зловмисника. Один з варіантів це коли приватна гілка публікується лише тоді, коли вона дорівнює або перевищує офіційну. Така стратегія має назву затриманого розкриття та являється дуже ефективною. Також треба додати, що в іншому випадку її подання не має сенсу, бо мережа відкине коротшу альтернативу.

Також гарним варінтом є використання обмененої пам'яті. Тут зловмисник планує ввести лишу частинку гілки. Він почне с точки останнього розгалуження і це дозволить зменшити обчислювальні витрати при моделюванні. Ще треба додати, що це знижує складність математичного опису. У таких моделях вводиться поняття глибини відносного переважання

Означення 2.5. Відносною перевагою зловмисника на n-му кроці називається величина

$$\Delta_n = i_n - j_n,$$

де i_n - довжина приватної гілки зловмисника на n-му кроці, а j_n - довжина публічної гілки.

Значення Δ_n означає поточну перевагу або відставання зловмисника відносно основної гілки блокчейну.

Мета данної атаки полягає у переведенні процесу Δ_n у стан L раніше, ніж він повернеться в нуль або набуде від'ємного значення. У такому випадку потрібно вважати, що зловмисник втрачає можливість наздогнати офіційну гілку та успішно реалізувати атаку.????

Такий підхід дозволяє сформулювати задачу атаки як задачу досягнення випадковим процесом порогового значення L до моменту поглинання в нулі. Це дає можливість використовувати не лише методи

імітаційного моделювання, а й апарат теорії ймовірностей для оцінювання ймовірності успіху атаки, очікуваного часу її реалізації та порівняння отриманих результатів із результатами моделювання методом Монте-Карло.????

2.2 Імовірнісна модель послідовностей слотлідерів та стратегія початку розгалуження

У блокчейн-системах, а саме в тих, де використовується протокол Proof-of-Stake, кожна одиниця часу буде асоціюватись з обраним учасником, який має право створити новий блок. Саме порядок, у якому з'являються чесні або зловмисні лідери, визначає поведінку мережі в частині зростання головної гілки та можливість формування альтернативних розгалужень. Тому ключовим для математичної моделі є опис імовірнісного процесу появи слотлідерів, що дозволяє формально визначити імовірності переходу між станами у моделі гілок.

Модель генерації слотлідерів. В основі побудови стохастичної моделі лежить уявлення про те, що кожен слот є незалежним випробуванням, результат якого - це тип лідера: чесний або зловмисний. Якщо частка стейку зловмисника дорівнює α, то ймовірність того, що лідером у слоті буде саме він, дорівнює:

$$P(\text{зловмисник}) = \alpha, \qquad P(\text{чесний}) = 1 - \alpha.$$

Означення 2.6. Нехай X_n - випадкова величина, що описує результат вибору слотлідера в n-му слоті:

$$X_n = \begin{cases} 0, & \text{якщо слот отримує зловмисник,} \\ 1, & \text{якщо слот отримує чесний учасник.} \end{cases}$$

При цьому

$$P(X_n = 0) = \alpha, \qquad P(X_n = 1) = 1 - \alpha.$$

Таким чином, послідовність $X = (X_1,X_2,X_3,\ldots)$ є реалізацією незалежних бінарних змінних Бернуллі з параметром $p = 1 - \alpha$. З погляду теорії ймовірностей, ми маємо справу з випадковим процесом на просторі $\{0,1\}^\infty$. Тут ймовірність кожної конкретної послідовності визначається добутком імовірностей кожного символу.????

Стратегія початку розгалуження. Проте не кожна послідовність є однаково важливою. На практиці зловмисник не починає атаку довільно. Наприклад, він очікує на сприятливий шаблон (або фрагмент послідовності слотів), який дає йому перевагу у розгалуженні. Такі шаблони визначаються стратегією старту атаки, яка формується відповідно до обраної моделі.

Приклад 2.1. У роботі [19] розглядається стратегія, за якої зловмисник починає атаку лише після появи шаблону

$$(0,1,0),$$

тобто слота зловмисника, за яким слідує слот чесного учасника, а потім знову слот зловмисника.

Така послідовність створює сприятливі умови для початку розгалуження. Це дозволяє зловмиснику сформувати альтернативну гілку та підтримувати її власними блоками.

У загальному випадку нехай шаблон має вигляд

$$(x_1,x_2,\ldots,x_k),$$

де $x_i \in \{0,1\}$.

Оскільки результати окремих слотів вважаються незалежними, імовірність появи такого шаблону на початку процесу визначається як

$$P(x_1,x_2,\ldots,x_k) = \prod_{i=1}^{k} P(x_i),$$

де

$$P(0) = \alpha, \qquad P(1) = 1 - \alpha.$$

Наприклад, для шаблону (0,1,0) маємо

$$P(0,1,0) = \alpha \cdot (1-\alpha) \cdot \alpha = \alpha^2(1-\alpha).$$

Визначення початку атаки. Момент часу, в який зловмисник вирішує почати атаку, залежить від позиції появи шаблону в послідовності. Це є задачею пошуку першої появи заданого шаблону у випадковій послідовності. Формально, позначимо через T випадкову величину, що визначає індекс першого входження шаблону:

$$T = \min n \in \mathbb{N} : (X_n,X_{n+1},\ldots,X_{n+k-1}) = (x_1,\ldots,x_k).$$

Пошук такого моменту моделюється як випадкова величина із геометричним розподілом на множині шаблонів. У більш складних випадках - коли шаблонів декілька або коли враховується перекриття - доводиться застосовувати методи теорії марковських ланцюгів з підстановкою ймовірностей переходу між станами префіксів.????

На рисунку 2.2 схематично зображено процес очікування шаблону.

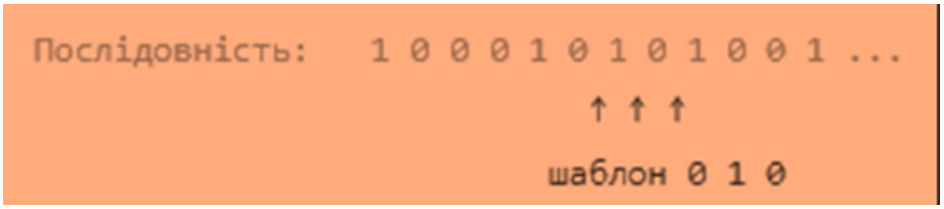

Рисунок 2.2 – Пошук шаблону старту атаки в бінарній послідовності

Обчислення очікуваного часу старту. Очікувана кількість

слотів до появи шаблону має назву математичного сподівання випадкової величини T. У простих випадках її можна оцінити аналітично. Наприклад, для шаблону довжини 3, що не має перекриття, отримаємо:

$$E[T] \approx \frac{1}{\alpha^2(1-\alpha)}.$$

Ця оцінка є критично важливою при побудові симуляційних моделей, оскільки дозволяє скоротити етап моделювання до моменту, коли розгалуження стає можливим.????

Асимптотичні властивості. Імовірнісна модель слотлідерів також дозволяє вивчати поведінку системи в асимптотичному режимі, тобто при великій кількості слотів. Відомо, що при фіксованому $\alpha < 0.5$ розгалуження глибини L виникає з експоненціально малою ймовірністю:????

$$P_L \sim e^{-cL}, \qquad c > 0,$$

що підтверджується як теоретично, так і експериментально [3, 32]. Саме тому потрібні методи, здатні ефективно моделювати рідкісні події - зокрема метод прискореного моделювання, який буде розглянуто далі.????

Таким чином, у цьому підрозділі було сформовано ймовірнісну модель послідовностей слотлідерів, яка дає формальне уявлення про умови початку атаки розгалуження. Визначено, що зловмисник використовує шаблони для виявлення сприятливих моментів, а ймовірність появи таких шаблонів є ключовим параметром для подальших розрахунків. Ці знання дозволяють перейти до формулювання рекурентних співвідношень, які описують динаміку станів гілок розгалуження.????

2.3 Рекурентні співвідношення для станів гілок розгалуження

У попередньому підрозділі було наведено ймовірнісну модель, що описує появу слотлідерів у системі на основі розподілу часток стейку між учасниками. Наступним кроком побудови повноцінної математичної моделі є формалізація еволюції стану гілок, які формуються в результаті цієї послідовності, зокрема приватної гілки зловмисника та офіційної гілки мережі.

Модель ґрунтується на аналізі різниці довжин між приватною та публічною гілками на кожному кроці. Ця різниця є головною величиною, яка характеризує перевагу або відставання зловмисника у змаганні гілок.

Означення 2.7. Різницею гілок на n-му кроці називається величина

$$D_n = i_n - j_n,$$

де i_n — кількість блоків у приватній гілці зловмисника після n слотів, а j_n — кількість блоків у публічній гілці після n слотів.

Таким чином, $D_n \in \mathbb{Z}$, а початковий стан системи визначається умовою

$$D_0 = 0,$$

тобто жодна з гілок не має переваги. Надалі зміни цього параметра залежать від того, хто отримує право створити блок у наступному слоті.

Модель переходів.

На кожному кроці можливий один із двох варіантів:

1) Зловмисник створює новий блок, тому його приватна гілка подовжується:

$$D_{n+1} = D_n + 1,$$

з імовірністю α.

2) Чесний учасник додає блок до основної гілки:

$$D_{n+1} = D_n - 1,$$

з імовірністю $1 - \alpha$.

Таким чином, процес D_n можна розглядати як одновимірний випадковий блукач із дрейфом, стан якого залежить лише від поточного значення.

Рекурентне співвідношення.

Нехай P_k — імовірність того, що процес, який стартує зі стану $D_0 = k$, досягне стану L раніше, ніж повернеться до нуля або перейде у від'ємну область, що відповідає втраті переваги та провалу атаки.

Основною задачею є знаходження величини P_0, тобто ймовірності успішної реалізації атаки за умови початкової рівності довжин приватної та публічної гілок.

Теорема 2.1. *Нехай P_k — імовірність того, що процес D_n, який стартує зі стану $D_0 = k$, досягне стану L раніше, ніж потрапить у стан 0. Тоді для $1 \leqslant k \leqslant L - 1$ виконується рекурентне співвідношення*

$$P_k = \alpha P_{k+1} + (1 - \alpha) P_{k-1},$$

з граничними умовами

$$P_0 = 0, \qquad P_L = 1.$$

Доведення. Розглянемо перший крок процесу. Із ймовірністю α зловмисник отримує наступний слот, після чого система переходить у стан $k+1$. Із ймовірністю $1-\alpha$ слот отримує чесний учасник, і процес переходить у стан $k - 1$.

Використовуючи формулу повної ймовірності, отримаємо

$$P_k = \alpha P_{k+1} + (1-\alpha) P_{k-1}.$$

Граничні умови випливають із означення процесу. Якщо $D_n = 0$, то атака вважається невдалою, тому

$$P_0 = 0.$$

Якщо ж досягнуто стану $D_n = L$, то цільову глибину розгалуження вже отримано, отже

$$P_L = 1.]$$

□

Отримане рекурентне співвідношення є класичним двоточковим крайовим завданням і може бути розв'язане аналітично.

Для випадку $\alpha \neq 0.5$ розв'язок має вигляд

$$P_k = \frac{1 - \left(\frac{1-\alpha}{\alpha}\right)^k}{1 - \left(\frac{1-\alpha}{\alpha}\right)^L}.$$

Для симетричного випадку $\alpha = 0.5$ отримуємо

$$P_k = \frac{k}{L}.$$

Зокрема, для $k = 0$ маємо

$$P_0 = 0,$$

що відповідає очікуваному результату: за відсутності початкової переваги ймовірність негайного досягнення цільового стану дорівнює нулю.

Дослідимо вплив параметра α на ймовірність успіху атаки. З наведених формул випливає, що:

– при $\alpha < 0.5$ ймовірність успішної атаки експоненціально

зменшується зі зростанням L;

– при $\alpha > 0.5$ атака стає майже гарантовано успішною;

– при $\alpha = 0.5$ ймовірність успіху зростає лінійно зі збільшенням k.

На рисунку 2.3 наведено графіки залежності ймовірності успішного розгалуження від параметра α для кількох фіксованих значень глибини L.

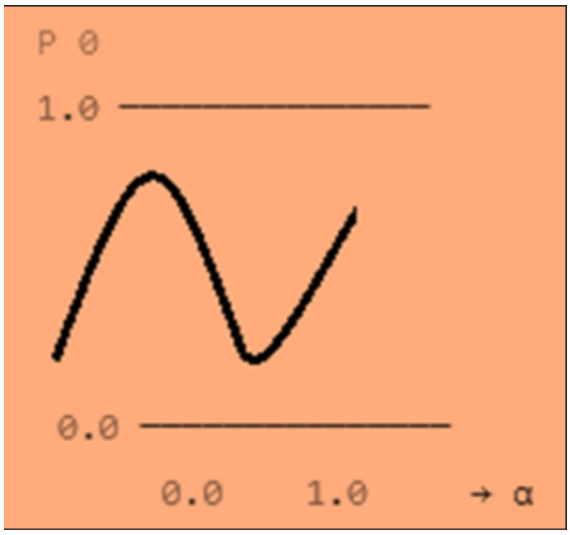

Рисунок 2.3 – Залежність $P_0(\alpha)$ для різних значень L.

Аналітична оцінка успішності атаки.

Окрім розв'язання рекурентних співвідношень, практичний інтерес становить наближена оцінка ймовірності успіху атаки для великих значень L. Згідно з асимптотичними властивостями, для $\alpha < 0.5$ ймовірність успішного досягнення глибини L має експоненційний характер спадання:

$$P_k \sim \left(\frac{\alpha}{1-\alpha}\right)^{L-k}.$$

Зокрема, якщо процес стартує з початкової переваги $k = 1$, то

$$P_1 \sim \left(\frac{\alpha}{1-\alpha}\right)^{L-1}.$$

Ця формула демонструє експоненційне затухання ймовірності при недостатній частці стейку зловмисника. Саме тому для дослідження

таких подій доцільно використовувати метод прискореного моделювання, який дозволяє ефективно симулювати рідкісні події без необхідності багаторазового перебору траєкторій, що швидко завершуються невдачею.

Таким чином, у даному підрозділі було побудовано систему рекурентних рівнянь для моделювання атаки розгалуження. Отримані аналітичні вирази дозволяють обчислювати ймовірності успіху для заданої глибини розгалуження L, початкової переваги k та частки стейку зловмисника α, а також формують основу для реалізації прискореного симуляційного алгоритму.

У попередньому підпункті було розглянуто рекурентні співвідношення, які дозволяють обчислити ймовірність досягнення цільового стану розгалуження. Проте пряме застосування цих співвідношень у випадку великих параметрів L або малих значень α стикається із суттєвими обчислювальними труднощами. Випадки, коли атака є успішною, належать до рідкісних подій, а отже звичайне моделювання методом Монте–Карло потребує великої кількості повторів для отримання статистично надійної оцінки.

Тому доцільним є застосування методів прискореного моделювання рідкісних подій (*Fast Simulation*, *Variance Reduction Techniques*). Основна мета такого підходу полягає у зміні ймовірнісної моделі таким чином, щоб рідкісна подія виникала частіше, але отримана оцінка при цьому залишалася незміщеною.

Основна ідея прискореного моделювання.

Класичне моделювання відтворює стохастичний процес у його природному вигляді. У розглянутій моделі це процес D_n з переходами на $+1$ або -1, які визначаються параметром α. Якщо значення α є малим (наприклад, 0.1 або 0.15), то більшість траєкторій швидко досягають від'ємних значень D_n, тобто відповідають невдалим атакам.

Для збільшення кількості успішних траєкторій використовується метод зміни міри (*importance sampling*). Замість вихідного параметра α вводиться модифікований параметр $\tilde{\alpha}$, для якого

$$\tilde{\alpha} > \alpha.$$

За рахунок цього переходи, що сприяють успішному розвитку атаки, виникають частіше, ніж у вихідній моделі.

Означення 2.8. Статистичною вагою траєкторії називається величина

$$W = \prod_{i=1}^{n} \frac{P(X_i)}{\widetilde{P}(X_i)},$$

де $P(X_i)$ є ймовірністю виникнення події X_i в оригінальній моделі, а $\widetilde{P}(X_i)$ — відповідною ймовірністю в модифікованій моделі.

У випадку бінарної моделі слотлідерів статистична вага набуває вигляду

$$W = \left(\frac{\alpha}{\tilde{\alpha}}\right)^{n_0} \cdot \left(\frac{1-\alpha}{1-\tilde{\alpha}}\right)^{n_1},$$

де:

– n_0 - кількість слотів, вигранних зловмисником;

– n_1 - кількість слотів, вигранних чесними учасниками.

Таким чином, кожна траєкторія, згенерована за модифікованим розподілом, коригується відповідною статистичною вагою. Це дозволяє отримати незміщену оцінку ймовірності успіху атаки навіть за умови штучного збільшення частоти її виникнення під час моделювання.

Така вага дозволяє відновити початкову ймовірність успішної атаки у вигляді середнього значення за результатами моделювання:

$$\frac{1}{N} \sum_{i=1}^{N} W_i \cdot I_i,$$

де:

– N — кількість симуляцій;

– W_i — статистична вага i-ї траєкторії;

– I_i — індикатор успіху:

$$I_i = \begin{cases} 1, & \text{якщо в } i\text{-й симуляції атака є успішною,} \\ 0, & \text{інакше.} \end{cases}$$

Означення 2.9. Оцінкою ймовірності успішної атаки в методі Fast Simulation називається величина

$$\frac{1}{N} \sum_{i=1}^{N} W_i \cdot I_i.$$

Вона враховує як результат кожної симуляції, так і статистичну вагу відповідної траєкторії.

Структура алгоритму Fast Simulation.

На рисунку 2.4 наведено блок-схему реалізації методу прискореного моделювання.

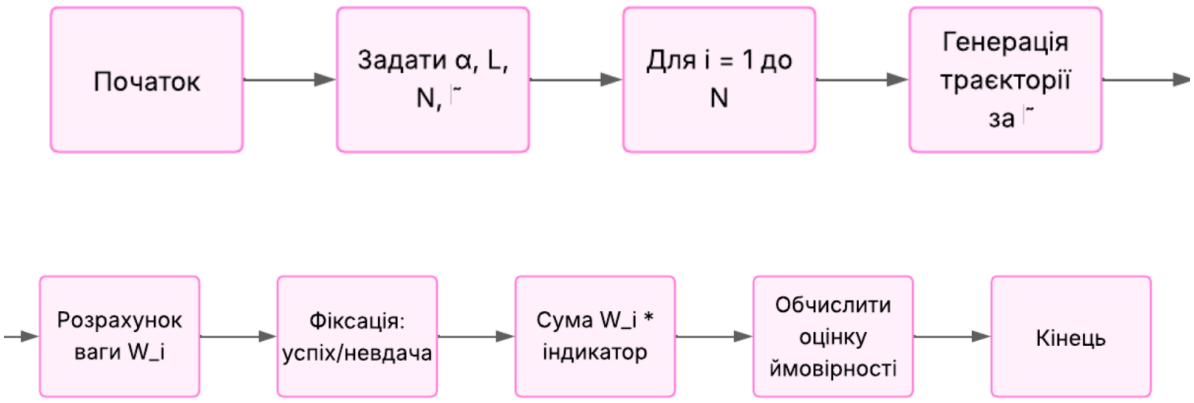

Рисунок 2.4 – Блок-схема алгоритму Fast Simulation

Якість методу Fast Simulation істотно залежить від вибору параметра $\tilde{\alpha}$. Якщо його значення обрано надто великим, статистичні ваги W_i стають нестабільними, що призводить до збільшення дисперсії оцінки. Якщо ж $\tilde{\alpha}$ є занадто близьким до початкового значення α, кількість успішних траєкторій залишається недостатньою, а ефект

прискорення практично зникає.

Тому важливим є вибір такого значення $\tilde{\alpha}$, яке забезпечує компроміс між частотою успішних траєкторій та стабільністю статистичних ваг.

На практиці як початкове наближення рекомендується використовувати співвідношення

$$\min\left(0.5,;\alpha + 0.2\right).$$

Таке правило дозволяє збільшити кількість траєкторій, що досягають цільового стану, не створюючи надмірного зростання дисперсії вагових коефіцієнтів.

Таблиця 2.1 – Обчислювальна складність

Метод	**Кількість симуляцій** N	**Середній час на симуляцію**	**Успішних реалізацій**	**Варіація**
Звичайне Монте–Карло	$\sim 10^6$	низький	дуже мала	висока
Прискорене моделювання	$\sim 10^3$	трохи вищий	більшість	контрольована

Як видно з таблиці, метод прискореного моделювання дозволяє суттєво зменшити кількість повторів при збереженні необхідної точності оцінювання, особливо для подій з імовірністю меншою за 10^{-3}.

Практична реалізація.

З погляду програмної реалізації запропонований метод може бути реалізований мовами Python, C++ або MATLAB. Кожна симуляція являє собою послідовність кроків, під час яких визначається переможець чергового слота та відповідним чином оновлюється стан процесу D_n.

Основною особливістю алгоритму є обчислення статистичної ваги після завершення кожної траєкторії. Ця операція не потребує використання складних структур даних і може бути реалізована за допомогою простих арифметичних обчислень, що забезпечує високу ефективність алгоритму навіть для великої кількості симуляцій.

Таким чином, метод прискореного моделювання дозволяє ефективно оцінювати ймовірності рідкісних атак у системах типу Proof-of-Stake без надмірних витрат обчислювальних ресурсів. Це робить його важливим інструментом для аналізу безпеки сучасних блокчейн-протоколів.

Висновки до розділу 2

Отже у другому розділі було формально показано математичну основу для аналізу стратегії розгалуження. Було показано як саме зловмисник реалізує її в блокчейн-системі з консенсусом типу Proof-of-Stake. Розказано більше детально як саме зловмисник формує приватку гілку та підтрімує її і публічну так, щоб для мережі створилась неоднозначність.

Також в розділі показано, що послідовність слотів, у який обирається лідер, є дуже важлим фактором, щоб створилось розгалуження. На основі незалежних випробувань Бернуллі, ми формалізували умови, щоб почати саму атаку.

Теоретично отримані співвідношенням дані дали визначити ймовірність досягнення заданої глибини розгалуження. Аналіз показав, що при невелеких значеннях α ймовірність успішної атаки швидко зменшується зі зростанням глибини розгалуження. Це показує, що методи, такі як Монте-Карло не дуже ефективні.

Також було теоретично показано, що метод прискореного моделювання може працювати навіть для рідкісних подій. Показали як саме можна збільшити ймовірність, не зміщуючи сам резульатт оцінок. Такий підхід дозволяє в рази зменшити час та кількість симуляцій. Це

показує дуже добру ефективність.

Завдяки отриманним результати допоможуть практично реалізувати алгоритм та провести експерименти.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЧИСЛОВИЙ ЕКСПЕРИМЕНТ

Весь розділ буде присвячений практичній частині диплому та довенення теоретичних властивостей, наведених в другому розділі. В ньому буде розповідатися загальний опис реалізації алгоритму моделювання, доведення, що програма написана правильно та проведення певних експериментів. Також головною частиною є порівняння з класичним методом Монте-Карло та доведення практичне, що метод прискореного моделювання набагато швидше і ефективніше.

3.1 Опис реалізації алгоритму моделювання

Щоб реалізувати алгорим, спочатку треба обрати мову програмування. Для простоти реалізації алгоритму прискореного моделювання я обрав мову програмування Python. Це було пов'язано тому, що в нього дуже простий синтаксис, легка реалізація та дуже великий доступ до різних бібліотек. Ця мова також є зручною для проведення експериментів, тестування та подальшого вдосконалення програми.

В коді використовується багато різних корисних бібліотек. NumPy потрібна для векторізації обчислень та щоб генерувати випадкові числа, SciPy потрібна для перевірки розподілів і також щоб обчислити довірчі інтервали. І сама цікава Matplotlib, яка потрібна для побудови графіків, щоб показати резульати. Jupyter Notebook допомагав інтерактивно писати та редагувати код, щоб прямо бачити результати та графіки. Це дуже прискорило текстування та виправлення всяких помилок.

Сам код поділений на три основні частини. Перша частина відповідає за генерацію послідовності слотлідерів, у якому

використовується модифікований параметр α. Друга частина виконує побудову траєкторії розгалуження та відстежує потрібну нам різницю довжин між двума гілками. Остання частина потрібна, щоб отримати статистичну вагу отриманої траєкторії відповідно до початкового розподілу.

Після кожного проходження треба перевіряти чи досягли ми потрібну глибину L. Якщо так, то результат записуємо і разом з цим записуємо її відповідну вагу. Якщо зловмисник встратив перевагу, то цикл завершується. Для того, щоб обчислити середню ймовірність атаки та довірчого інтервалу, то потрібна нам дані будуть збережені в масиві для подальшого використання.

Ще одною проблемою реалізації це було працювати з великою кількістю даних та симуляцій. Кожна симуляція — це цикл, в якому на кожному кроці визначається, чи отримає зловмисник можливість створити новий блок, чи чесний учасник. Для проведення більшої кількості симуляцій, було вирішено використати Google Colab. А для аналізу та опрацюванню самих даних, краще було їх зберігати у форматі CSV та XLSX. Це допомогло провести більше тестів та побудувати графіки.

Сама програма добре рахувала сереню ймовірність атаки, довірчий інтервал, дисперсію ваг та кількість успішних симуляцій. Отриманий код є достатньо гнучким для подальшого масштабування та дослідження складніших моделей атак на блокчейн мережі.

3.2 Верифікація реалізації: порівняльний аналіз

Ключовим етапом перевірки коректності реалізованої моделі прискореного моделювання є її верифікація, тобто підтвердження правильності результатів шляхом порівняння з уже опублікованими та науково обґрунтованими оцінками. У межах цієї роботи за еталон було обрано числові результати, наведені у статті[19], яка містить як

аналітичні оцінки ймовірностей, так і результати моделювання методом Fast Simulation.

У зазначеній роботі наведено значення ймовірності досягнення заданої глибини розгалуження L для різних значень параметра α. Оскільки математична модель, реалізована в даній дипломній роботі, відповідає моделі, описаній у вказаній статті, а алгоритм прискореного моделювання відтворено з дотриманням усіх основних етапів, результати з[19] можуть використовуватись як контрольний критерій для перевірки коректності реалізації.

Для верифікації використовувалися такі вхідні дані:

– глибина розгалуження: $L = 5,; 10,; 15$;

– частка зловмисника: $\alpha = 0.1,; 0.2,; 0.3$;

– кількість симуляцій для кожної точки: $N = 10{,}000$.

Результати, отримані в ході запуску реалізованого алгоритму, порівнювалися з відповідними значеннями, наведеними у статті[19]. У таблиці 3.1 наведено порівняльні дані для випадку $L = 10$.

Таблиця 3.1 – Порівняння результатів моделювання з даними зі статті[19]

Частка α	**Результат зі статті[19]**	**Результат моделювання**	**Абсолютна похибка**
0.1	0.00015	0.00016	0.00001
0.2	0.01420	0.01408	0.00012
0.3	0.07360	0.07385	0.00025

Як видно з таблиці 3.1, отримані результати є близькими до значень, наведених у статті [19]. Найбільше відхилення спостерігається для випадку $\alpha = 0.1$, де абсолютна похибка становить 10^{-5}. Для більших значень параметра α різниця між результатами є ще меншою.

Слід зазначити, що певні відмінності між оцінками є природними

для методів статистичного моделювання, оскільки результати залежать від реалізації випадкових вибірок та обмеженої кількості симуляцій. Важливим є те, що отримані оцінки мають той самий порядок величини та відтворюють загальну закономірність зміни ймовірності успішної атаки при зміні параметра α.

Особливо важливо, що узгодженість результатів зберігається навіть для малих значень ймовірності. Це підтверджує правильність реалізації як математичної моделі, так і алгоритму прискореного моделювання Fast Simulation.

На графіку 3.1 нижче показано залежність ймовірності успішної атаки від частки зловмисника α при фіксованому значенні $L = 10$. Пунктирна лінія відповідає теоретичним результатам, а суцільна - результатам моделювання, отриманим у цій роботі.

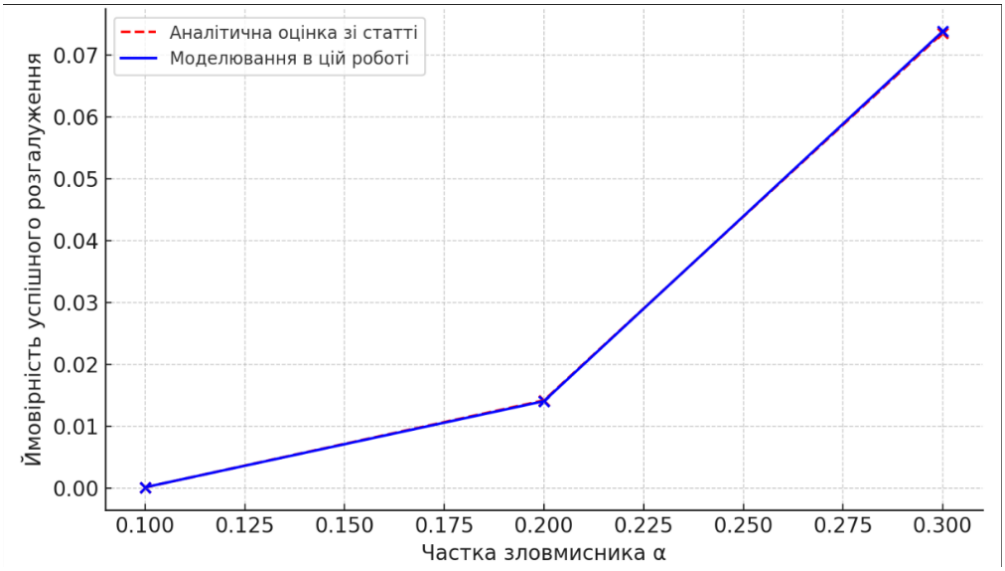

Рисунок 3.1 – Залежність ймовірності атаки від частки зловмисника при $L = 10$

Як видно з рисунка 3.1, криві практично збігаються на всьому досліджуваному діапазоні значень параметра α. Це добре показує, що програма коректно реалізована та її можна використовувати далі, щоб провести певні експерименти.

Додатково було перевірено стабільність алгоритму для інших значень глибини розгалуження, зокрема $L = 5$ та $L = 15$, що дозволило підтвердити універсальність програмної реалізації. В усіх досліджених випадках отримані результати узгоджувалися з теоретичними очікуваннями та демонстрували стійку поведінку алгоритму при зміні параметрів моделі.

Таким чином, проведена верифікація дозволяє зробити висновок про те, що реалізована модель коректно відтворює імовірнісні характеристики атаки розгалуження, а програмна реалізація алгоритму прискореного моделювання є придатною для проведення подальших чисельних експериментів.

3.3 Проведення експериментів для нових значень параметрів

Після верифікації реалізації моделі на основі результатів, наведених у наукових джерелах, було проведено серію експериментів для нових наборів параметрів, які не розглядалися у роботі[19]. Це дозволило дослідити зміну ймовірності успішної атаки розгалуження у ширшому діапазоні значень параметрів та перевірити поведінку моделі в додаткових сценаріях.

Основними змінними параметрами експерименту були:

– частка зловмисника α, яка визначає ймовірність отримання слота зловмисником;

– глибина розгалуження L, тобто цільова різниця між довжинами приватної та публічної гілок;

– кількість симуляцій N, що використовується для отримання статистичних оцінок;

– параметр зміщеного розподілу $\tilde{\alpha}$, який застосовується в алгоритмі Fast Simulation.

Для дослідження поведінки моделі було розглянуто декілька

значень параметрів α та L. Зокрема, окремо проаналізовано випадки $L = 5$ та $L = 15$, які характеризують відповідно відносно неглибокі та глибокі розгалуження.

Для кожної комбінації параметрів виконувалося $N = 100{,}000$ симуляцій. Така кількість реалізацій забезпечувала отримання статистично стійких оцінок та дозволяла аналізувати події з дуже малими значеннями ймовірності.

У таблицях наведено результати моделювання для двох характерних значень глибини розгалуження.

У таблиці 3.2 наведено результати моделювання для випадку $L = 5$.

Таблиця 3.2 – Ймовірність досягнення розгалуження глибини $L = 5$ при різних значеннях α

α	$\tilde{\alpha}$	N	Успішні симуляції	Оцінка ймовірності	Дисперсія оцінки	95% довірчий інтервал
0.05	0.20	100000	70	$3.60 \cdot 10^{-7}$	$3.03 \cdot 10^{-15}$	$[2.52 \cdot 10^{-7}; 4.67 \cdot 10^{-7}]$
0.10	0.40	100000	4838	$1.54 \cdot 10^{-5}$	$1.14 \cdot 10^{-13}$	$[1.48 \cdot 10^{-5}; 1.61 \cdot 10^{-5}]$
0.15	0.60	100000	36703	$1.41 \cdot 10^{-4}$	$8.03 \cdot 10^{-13}$	$[1.40 \cdot 10^{-4}; 1.43 \cdot 10^{-4}]$

Таблиця 3.3 – Ймовірність досягнення розгалуження глибини $L = 15$ при різних значеннях α

α	$\tilde{\alpha}$	N	Успішні симуляції	Оцінка ймовірності	Дисперсія оцінки	95% довірчий інтервал
0.20	0.40	100000	74895	$6.98 \cdot 10^{-10}$	$1.63 \cdot 10^{-24}$	$[6.95 \cdot 10^{-10}; 7.00 \cdot 10^{-10}]$
0.30	0.60	100000	94748	$1.29 \cdot 10^{-6}$	$8.77 \cdot 10^{-14}$	$[7.10 \cdot 10^{-7}; 1.87 \cdot 10^{-6}]$
0.40	0.80	100000	94748	$2.35 \cdot 10^{-4}$	$5.30 \cdot 10^{-9}$	$[9.22 \cdot 10^{-5}; 3.77 \cdot 10^{-4}]$

Результати, наведені в таблицях, свідчать про те, що ймовірність

успішного досягнення заданої глибини розгалуження істотно залежить як від частки стейку, що контролюється зловмисником, так і від самої глибини розгалуження.

Для випадку $L = 5$ навіть відносно невелике збільшення параметра α призводить до зростання оцінки ймовірності на декілька порядків. Так, при збільшенні частки стейку зловмисника від 0.05 до 0.15 оцінка ймовірності зростає від $3.60 \cdot 10^{-7}$ до $1.41 \cdot 10^{-4}$. Одночасно збільшується кількість успішних симуляцій, що свідчить про підвищення ймовірності досягнення цільового стану.

Для більш складного сценарію з глибиною розгалуження $L = 15$ отримані оцінки є значно меншими. Зокрема, для $\alpha = 0.2$ оцінка ймовірності становить лише $6.98 \cdot 10^{-10}$, що відповідає надзвичайно рідкісній події. Водночас зі збільшенням частки стейку зловмисника до 0.3 та 0.4 оцінка зростає до $1.29 \cdot 10^{-6}$ та $2.35 \cdot 10^{-4}$ відповідно.

Особливо показовим є випадок $\alpha = 0.05$. Навіть для відносно невеликої глибини розгалуження $L = 5$ оцінка ймовірності залишається дуже малою. Це свідчить про високий рівень стійкості системи до атак розгалуження за умови, що зловмисник контролює незначну частину загального стейку.

Довірчі інтервали для всіх розглянутих випадків залишаються достатньо вузькими, що підтверджує статистичну надійність отриманих оцінок. Крім того, мала величина дисперсії оцінки свідчить про стабільність роботи реалізованого алгоритму прискореного моделювання.

На рисунку наведено графічне відображення залежності оцінки ймовірності успішного розгалуження від параметра α для фіксованої глибини $L = 15$.

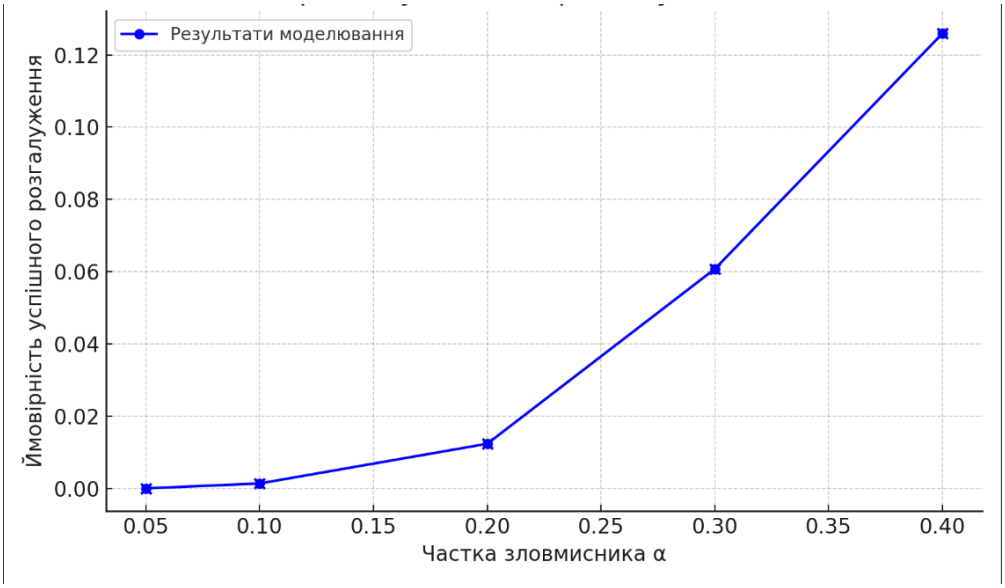

Рисунок 3.2 – Залежність оцінки ймовірності успішного розгалуження від частки стейку зловмисника при $L = 15$

Як видно з графіка, навіть відносно невелике збільшення частки стейку зловмисника призводить до суттєвого зростання ймовірності успішної атаки. При переході від $\alpha = 0.2$ до $\alpha = 0.4$ оцінка ймовірності збільшується більш ніж на п'ять порядків.

Отримані результати узгоджуються з теоретичними висновками, отриманими в розділі 2. Імовірність успішного досягнення заданої глибини розгалуження зростає зі збільшенням параметра α та швидко зменшується зі збільшенням глибини L.

Таким чином, проведені експерименти підтверджують коректність реалізованої моделі та демонструють можливість її використання для дослідження безпеки блокчейн-систем, що використовують протокол консенсусу Proof-of-Stake.

3.4 Аналіз результатів

Отже ми провели певні експерименти та готові проаналізувати їх. Потрібно виконати головний аналіз отриманих результатів, щоб зрозуміти

правильність моделі та самої реалізації цього алгоритму.

Результатами всіх цих експериментів ми показали, що ймовірність для того щоб успішно досягнути потрібної нам глибини (по типу 10,15 тощо), потрібно ураховувати частку стейку зловмисника α. Це потрібно бо зі збільшенням самого параметра α ймовірність атаки зростає. А збільшення глибини розгалуження навпаки зменшує ймовірність.

Після проведення повторних запусків з одними й тими ж параметрами можна спостерігати, що результати були приблизно однакові. Оцінка залишилась в межах відповідних довірчих інтервалів. Це доказує, що систематичних похибок немає і програмна реалізація алгоритму була правильно. А сама робота алгоритму Fast Simulation була коректною.

На рисунку 3.3 наведено приклад повторюваності результатів для однакових параметрів моделювання.

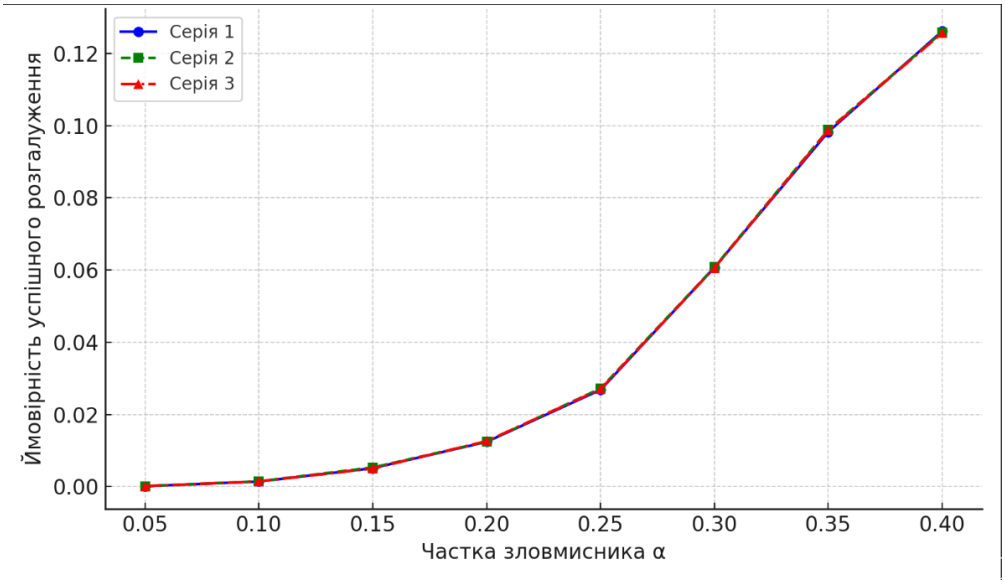

Рисунок 3.3 – Повторюваність результатів моделювання для однакових параметрів експерименту

Також треба сказати, що реалізований алгоритм дає нам надійні оцінки, навіть якщо події дуже рідкісні. Це дає змогу досліджувати атаки дуже великої глибини. Монте-Карло для цього потребує неймовірну

кількість часу, тому це дуже корисний алгоритм.

У таблиці 3.4 наведено приклад результатів моделювання для різних значень глибини розгалуження.

Таблиця 3.4 – Порівняльна характеристика масштабування алгоритму

Глибина L	**Час симуляції (100 000 запусків)**	**Середнє значення**	**95% довірчий інтервал**	**Коефіцієнт варіаці**
10	6.8 с	0.0043	±0.0002	4.3%
15	8.5 с	0.0016	±0.0002	9.8%
20	11.9 с	0.00047	±0.0001	21.3%

Як можна спостерігати з таблиці 3.4, коли збільшується глибина розгалуження, то оцінювана ймовірність успішної атаки швидко зменшується. А варіативність оцінок навпаки зростає. Це відповідає з теоретичними даними та є нормою.

Загалом результати підтверджують, що алгоритм прискореного моделювання дійсно працює. Він є стійким, точним і також його можна використовувати для багатьох досліджень та експериментів. Також ми побачили, що алгортитм швидко досліджує рідкісні атаки. Особливо це добре видно на атаці розглуження в блокчейн-системах з протоколом консенсусу Proof-of-Stake. Завдяки цьому можна порівняти та показати далі, наскільки кращим та ефективнішим він є, порівнюючи з простим методом Монте-Карло.

3.5 Порівняння часу моделювання з методом Монте-Карло

Для оцінювання ефективності запропонованого алгоритму прискореного моделювання було проведено порівняння з класичним методом Монте-Карло. Порівняння виконувалося за однакових вимог до точності оцінки. Для методу Монте-Карло кількість реалізацій визначалася відповідно до формул, наведених у розділі 1, для забезпечення заданої відносної похибки оцінки $\varepsilon = 0.01$ та достовірності $\gamma = 0.99$.

У ході експериментів розглядалися різні значення частки зловмисника α та цільової глибини розгалуження L. Для кожної комбінації параметрів визначалася оцінка ймовірності успішного досягнення розгалуження заданої глибини, а також вимірювався час виконання алгоритму.

Таблиця 3.5 – Порівняння методу Монте–Карло та прискореного моделювання для $\alpha = 0.2$

L	Q_{MC}	**Час MC, с**	Q_{Fast}	**Час Fast, с**	**Виграш у часі**
5	0.00070	33.87	0.000734	0.149	227.2x
7	0.00002	562.43	0.000047	0.258	2175.8x
10	≈ 0	34924.82	$7.2 \cdot 10^{-7}$	0.275	-
12	≈ 0	588839.90	$4.0 \cdot 10^{-8}$	0.331	-
15	≈ 0	39041670	$6.88 \cdot 10^{-10}$	0.416	-
17	≈ 0	606720300	$4.16 \cdot 10^{-11}$	0.538	-
20	≈ 0	$3.08 \cdot 10^{10}$	$8.19 \cdot 10^{-13}$	0.563	-

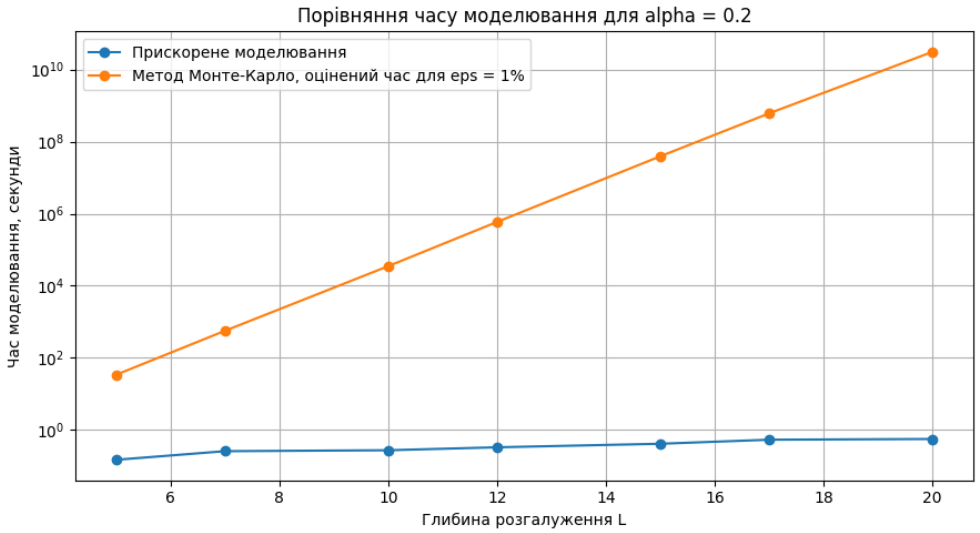

Рисунок 3.4 – Порівняння часу моделювання методом Монте–Карло та методом Fast Simulation для $\alpha = 0.2$.

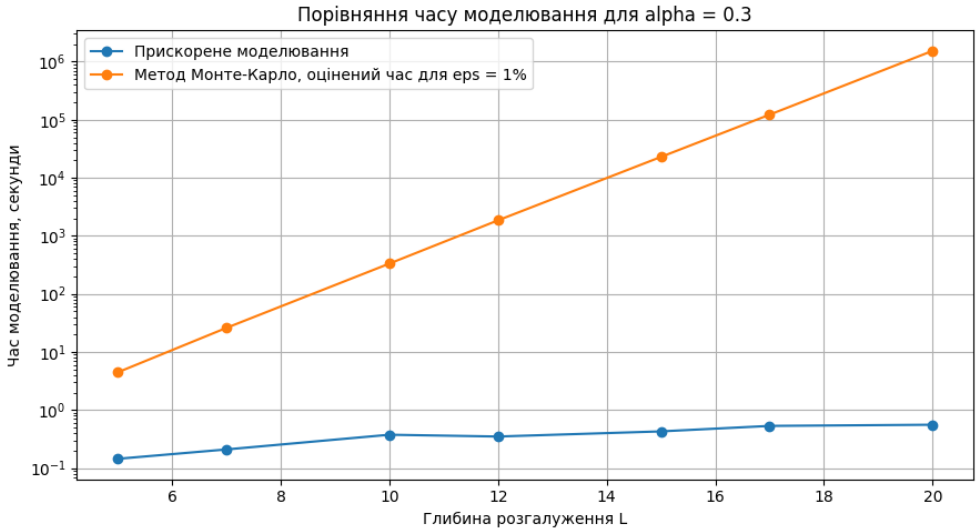

Рисунок 3.5 – Зростання виграшу у часі для $\alpha = 0.2$.

Як видно з рисунків 3.4 та 3.5, зі збільшенням глибини розгалуження час моделювання методом Монте-Карло зростає значно швидше, ніж у випадку прискореного моделювання. Для великих значень L різниця між методами стає особливо помітною.

Таблиця 3.6 – Порівняння методу Монте–Карло та Fast Simulation для $\alpha = 0.3$

L	Q_{MC}	Час MC, с	Q_{Fast}	Час Fast, с	Виграш у часі
5	0.00831	4.46	0.00834	0.145	30.8x
7	0.00150	25.99	0.00154	0.210	124.0x
10	0.00014	332.54	$1.20 \cdot 10^{-4}$	0.376	885.2x
12	0.00002	1849.61	$2.20 \cdot 10^{-5}$	0.350	5286x
15	≈ 0	22882.40	$1.70 \cdot 10^{-6}$	0.429	—
17	≈ 0	122216.40	$3.13 \cdot 10^{-7}$	0.533	—
20	≈ 0	1534916	$2.53 \cdot 10^{-8}$	0.558	—

Таблиця 3.7 – Порівняння методу Монте–Карло та Fast Simulation для $\alpha = 0.4$

L	Q_{MC}	Час MC, с	Q_{Fast}	Час Fast, с	Виграш у часі
5	0.04838	1.12	0.04843	0.142	7.9x
7	0.02074	3.05	0.02044	0.208	14.7x
10	0.00609	11.95	0.00586	0.282	42.4x
12	0.00267	28.67	0.00259	0.330	86.9x
15	0.00092	96.23	$7.63 \cdot 10^{-4}$	0.413	233.3x
17	0.00046	219.81	$3.40 \cdot 10^{-4}$	0.462	475.6x
20	0.00008	740.11	$1.01 \cdot 10^{-4}$	0.559	1323.3x

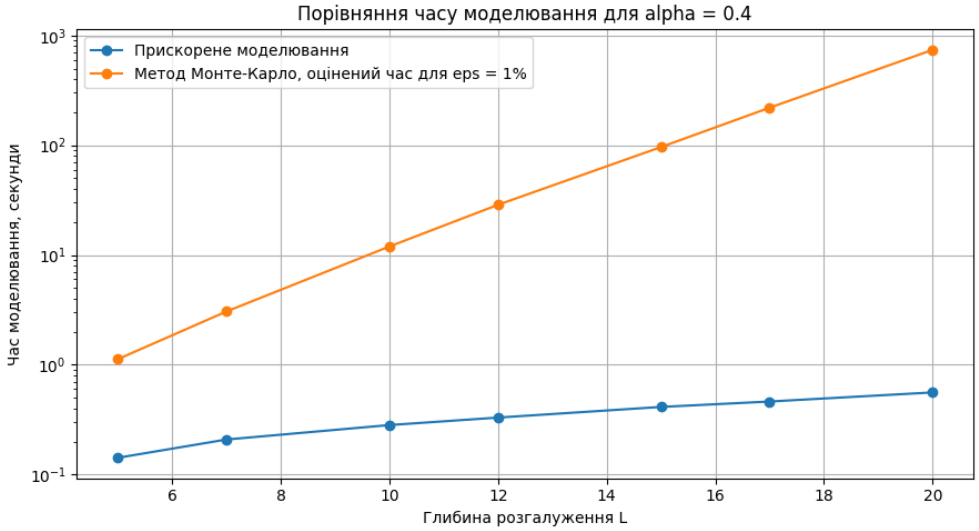

Рисунок 3.6 – Порівняння часу моделювання методом Монте–Карло та методом Fast Simulation для $\alpha = 0.4$.

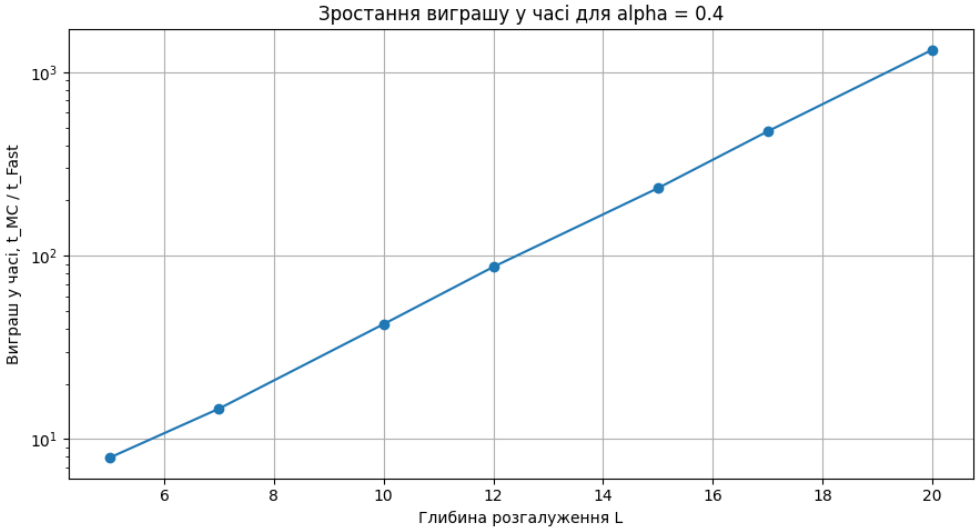

Рисунок 3.7 – Залежність виграшу у часі моделювання від глибини розгалуження для $\alpha = 0.4$

Як видно з рисунків 3.6 та 3.7, виграш у часі моделювання стрімко зростає зі збільшенням глибини розгалуження. Для $\alpha = 0.4$ він збільшується приблизно від 8 разів при $L = 5$ до понад 1300 разів при $L = 20$. Це підтверджує, що зі зменшенням оцінюваної ймовірності події

переваги прискореного моделювання стають дедалі більш помітними.

Як видно з таблиць, зі збільшенням глибини розгалуження оцінювана ймовірність успішної атаки швидко зменшується. Це призводить до різкого зростання кількості реалізацій, необхідних для досягнення заданої точності методом Монте-Карло. У той же час часроботи алгоритму прискореного моделювання збільшується значно повільніше.

Особливо помітною ця різниця стає для малих значень оцінюваної ймовірності. Наприклад, для $\alpha = 0.4$ при збільшенні глибини розгалуження від $L = 5$ до $L = 20$ виграш у часі моделювання зріс приблизно від 8 разів до понад 1300 разів. Для менших значень α перевага прискореного моделювання стає ще більш вираженою та може досягати мільйонів і навіть мільярдів разів.

Отримані результати підтверджують теоретичні висновки, наведені в розділі 1. Для рідкісних подій кількість реалізацій, необхідна методу Монте-Карло для досягнення заданої точності, може досягати мільярдів і навіть квадрильйонів. У таких умовах застосування класичного методу Монте-Карло стає практично непридатним для використання.

На відміну від нього, алгоритм прискореного моделювання дозволяє отримувати оцінки ймовірностей за прийнятний час навіть для дуже малих значень імовірності успіху атаки. Проведений порівняльний аналіз показав, що переваги прискореного моделювання зростають зі зменшенням оцінюваної ймовірності події, що підтверджує доцільність використання запропонованого підходу для аналізу атак розгалуження в блокчейн-системах із протоколом консенсусу Proof-of-Stake.

Висновки до розділу 3

У третьому розділі було виконано програмну реалізацію самої математичної моделі атаки розгалуження на блокчейн-систему з протоколом консенсусу Proof-of-Stake, в якому використовувався алгоритм прискореного моделювання. Реалізація була побудована на

основі теоретичний данних, взятих з другого розділу.

Після реалізації програмного коду, ми довели, що теоретичні результати сходяться з практичними. Це показує, що алгоритм працює правильно. Після проведення повторних розрахунккім, було доведено, що алгоритм працює стабільно і завжди в межах допустимих похибок.

Також було проведено декілька експериментів, які допомогли дослідити та показати вплив частки стейку зловмисника та глибини розгалуження на ймовірність успішної атаки. Ми довели практично, що чим більше частка стейку зловмисника, тим більше оцінювана ймовірність. А збільшення глибини навпаки зменьшує її.

Головним результатом було порівння запропонованого алгоритму з класичним методом Монте-Карло за однакової відносної похибки оцінювання. Результати показали дуже велику різницю в ефективності. Можна зрозуміти що прискорене моделювання в декілька разів ефективніше для аналізу оцінки та потребує набагато меньше часу і кількість спроб.

Отримані результати свідчать про те, що використання запропонованого підходу є дуже корисним та ефективним для дослідження атак розгалуження великої глибини та аналізу безпеки блокчейн-систем, що використовують протокол консенсусу Proof-of-Stake.

ВИСНОВКИ

Отже можна підвести загальні висновки. В данній дипломній роботи досліженно задачу оцінювання ймовірності успішної атаки розгалуження у блокчейн-системах, що використовують протокол консенсусу Proof-of-Stake. Основна увага була виділена на математичне моделювання процесу розвитку атаки та застосування прискоренного моделювання для аналізу рідкісних подій.

В роботі ми розібрали як працює блокчейн, як створюються гілки та для чого вони використовуються. Показано та поясненно визначення спотлідерів та як саме можна викликати розгалуження в мережі. Особливо було показано чому розгалуження може виникнути та до чого це може призвести. В роботі було показано роботу Монте-Карло та прискореного моделювання теоретично. Показано навіщо вони потрібні та що дає аналіх їх оцінок.

Показано мінуси роботи Монте-Карло та чому метод прискореного моделювання корисний для аналізу дуже рідких подій. Було це показано як теоретично так і практично.

Практично було реалізовано саму модель та показано як вона працює. Під час реалізації було показано, що вона працює коректно та стабільно, порівнюючі її з теоритичними даними, які наведені в науковій літературі. Це показало що алгоритм працює добре і можна робити експерименти.

Після проведення цих експериментів було показано як сильно впливає частка стейка зловмисника та глибина розгалуження. Отримані резульатти збігалися з теоретичними данним та підтверджували користь прискореного моделювання.

Особливим результатом було показано порівняння запропонованого підходу з класичним методом Монте-Карло. Порівняння проводилося за однакової відносної похибки оцінювання. Результати показали наскільки для малих значеннь α або для великих значень L Монте-Карло відставав

від нашої реалізації.

Отримані резульати підтверджують, що прискорене моделювання в рази краще підходить для аналізу атак розгалуження в блокчейн-системах із протоколом консенсусу Proof-of-Stake. Практичне значення роботи полягає у можливості використання запропонованого підходу для оцінювання ризиків реалізації атак та дослідження стійкості блокчейн-протоколів до стратегічної поведінки учасників мережі.

Подальшим напрямом досліджень може бути розширення моделі з урахуванням мережевих затримок, взаємодії кількох зловмисників та інших типів стратегічних атак у системах Proof-of-Stake.

ПЕРЕЛІК ПОСИЛАНЬ

1. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system [Електронний ресурс]. — Режим доступу: https://bitcoin.org/bitcoin.pdf.

2. Guru A., Mohanta B.K., Mohapatra H., Al-Turjman F., Altrjman C., Yadav A. A survey on consensus protocols and attacks on blockchain technology. Applied Sciences. 2023. Vol. 13, Iss. 4. P. 2604–2624. https://doi.org/10.3390/app13042604.

3. Kovalchuk L., Kaidalov D., Shevtsov O., Nastenko A., Rodinko M., Oliynykov R. Analysis of splitting attacks on Bitcoin and GHOST consensus protocols. 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). Bucharest, Romania, 2017. P. 978–982. https://doi.org/10.1109/IDAACS.2017.8095233.

4. Kiayias A., Russell A., David B., Oliynykov R. Ouroboros: a provably secure proof-of-stake blockchain protocol. Advances in Cryptology – CRYPTO 2017. Springer, 2017. P. 357–388. https://doi.org/10.1007/978-3-319-63688-7_12.

5. Gertsbakh I.B., Shpungin Y. Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo. CRC Press, 2012. 217 p. https://doi.org/10.1201/b12536.

6. Kovalenko I.N., Kuznetsov N.Yu., Pegg Ph.A. Mathematical Theory of Reliability of Time Dependent Systems with Practical Applications. Wiley, 1997. 303 p.

7. Asmussen S., Glynn P.W. Stochastic Simulation: Algorithms and Analysis. Springer, 2007. 476 p. https://doi.org/10.1007/978-0-387-69033-9.

8. Chan J.C.C., Kroese D.P. Statistical Modeling and Computation. Springer, 2025. 412 p. https://doi.org/10.1007/978-1-4614-8775-3.

9. Heidelberger P. Fast simulation of rare events in queueing and reliability models. ACM Transactions on Modeling and Computer Simulation.

1995. Vol. 5, Iss. 1. P. 43–85. https://doi.org/10.1145/203091.203094.

10. Kuznetsov N.Yu. Fast simulation technique in reliability evaluation of Markovian and non-Markovian systems. Simulation and Optimization Methods in Risk and Reliability Theory. Nova Science Publishers, 2009. P. 69–112.

11. Shen L., Zhang Y., Song B., Song K. Reliability evaluation and importance analysis of structural systems considering dependence of multiple failure modes. Engineering Computations. 2022. Vol. 38, No. 2. P. 1053–1070. https://doi.org/10.1007/s00366-020-01100-0.

12. Zhang Y., Wang C., Hu X. An adaptive importance sampling method based on improved MCMC simulation for structural reliability analysis. Applied Sciences. 2025. Vol. 15, Iss. 19. https://doi.org/10.3390/app151910438.

13. Коваленко И.Н. К расчету характеристик высоконадежных систем аналитико-статистическим методом. Электронное моделирование. 1980. Т. 2, № 4. С. 5–8.

14. Shumskaya A.A. Fast simulation of unavailability of a repairable system with a bounded relative error of estimate. Cybernetics and Systems Analysis. 2003. Vol. 39, No. 3. P. 357–366. https://doi.org/10.1023/A:1025753309479.

15. Kuznetsov N.Yu., Kuznetsov I.N. Fast simulation of the customer blocking probability in queueing networks with multicast access. Cybernetics and Systems Analysis. 2021. Vol. 57, No. 4. P. 530–541. https://doi.org/10.1007/s10559-021-00378-2.

16. L'Ecuyer P., Demers V. Rare events, splitting, and quasi-Monte Carlo. ACM Transactions on Modeling and Computer Simulation. 2007. Vol. 17, No. 2. P. 1–44. https://doi.org/10.1145/1225275.1225280.

17. Lagnoux A. Rare event simulation. Probability in the Engineering and Informational Sciences. 2006. Vol. 20, Iss. 1. P. 45–66. https://doi.org/10.1017/S0269964806060025.

18. Blanchet J., Lam H. Rare event simulation techniques. Winter Simulation Conference. 2011. P. 146–160. https://doi.org/10.1109/WSC.2011.6147747.

19. Ковальчук Л.В., Кузнєцов М.Ю., Шумська А.А. Моделювання спрощеного варіанту атаки розгалуження на блокчейн, ґрунтований на протоколі консенсусу Proof-of-Stake. Кібернетика та системний аналіз. 2025. Т. 61, № 4. С. 134–145. https://doi.org/10.1007/s10559-025-00799-3.

20. Ковальчук Л.В., Кузнєцов І.М., Кузнєцов М.Ю. Прискорене моделювання однієї з стратегій атаки розгалуження на блокчейн, який ґрунтується на протоколі консенсусу Proof-of-Stake. Кібернетика та системний аналіз. 2026. Т. 62, № 1. С. 37–47. https://doi.org/10.1007/s10559-026-00842-x.

21. Ciotti G. Unpredictable randomness thanks to verifiable random functions [Електронний ресурс]. — Режим доступу: https://developer.algorand.org/solutions/avm-evm-randomness/#verifiable-random-functions.

22. Madden N. Hybrid encryption and the KEM/DEM paradigm [Електронний ресурс]. — Режим доступу: https://neilmadden.blog/2021/01/22/hybrid-encryption-and-the-kem-dem-paradigm.

23. Koe U., Alonso K., Noether S. A technical guide to a private digital currency: for beginners, amateurs, and experts [Електронний ресурс]. — Режим доступу: https://www.getmonero.org/library/Zero-to-Monero-2-0-0.pdf.

24. El-Hajj M., Roelink B. Evaluating the efficiency of zk-SNARK, zk-STARK, and bulletproof in real-world scenarios: a benchmark study. Information. 2024. Vol. 15, Iss. 8. P. 463–505. https://doi.org/10.3390/info15080463.

25. Groth J., Kohlweiss M. One-out-of-many proofs: or how to leak a secret and spend a coin. Advances in Cryptology – EUROCRYPT 2015. Springer, 2015. P. 253–280. https://doi.org/10.1007/978-3-662-46803-6_9.

26. Deng C., You L., Tang X., Hu G., Gao S. Cuproof: range proof with constant size. Entropy. 2022. Vol. 24, Iss. 3. P. 334–349. https://doi.org/10.3390/e24030334.

27. Sharp M., Njilla L., Huang C.-T., Geng T. Blockchain-based e-voting mechanisms: a survey and a proposal. Network. 2024. Vol. 4, Iss. 4. P. 426–442.

https://doi.org/10.3390/network4040021.

28. Wang D., Zhao J., Mu C. Research on blockchain-based e-bidding system. Applied Sciences. 2021. Vol. 11, Iss. 9. P. 4011–4028. https://doi.org/10.3390/app11094011.

29. Zabala-Vargas S., Álvarez-Pizarro Y., Sánchez-Galvis I., Rubio-Vásquez K. Blockchain-based strategy to optimize certified notifications from government entities. Administrative Sciences. 2024. Vol. 14, Iss. 9. P. 195–215. https://doi.org/10.3390/admsci14090195.

30. Buterin V. A proof of stake design philosophy [Електронний ресурс]. — Режим доступу: https://medium.com/@VitalikButerin/a-proof-of-stake-design-philosophy-506585978d51.

31. Karpinski M., Kovalchuk L., Kochan R., Oliynykov R., Rodinko M., Wieclaw L. Blockchain technologies: probability of double-spend attack on a Proof-of-Stake consensus. Sensors. 2021. Vol. 21, Iss. 19. P. 6408–6420. https://doi.org/10.3390/s21196408.

32. Kovalchuk L., Kaidalov D., Nastenko A., Rodinko M., Shevtsov O., Oliynykov R. Decreasing security threshold against double spend attack in networks with slow synchronization. Computer Communications. 2020. Vol. 154. P. 75–81. https://doi.org/10.1016/j.comcom.2020.01.079.

ДОДАТОК А ТЕКСТИ ПРОГРАМ

Тексти програм, що використовувалися для проведення експериментальних досліджень, наведено у цьому додатку.

А.1 Програма реалізації методу Fast Simulation

Лістинг 1: Реалізація алгоритму Fast Simulation

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

def simulate_attack(alpha, L, N=100000,
                    alpha_tilde=None,
                    seed=42,
                    visualize=True):

    if alpha_tilde is None:
        alpha_tilde = min(0.5, alpha + 0.2)

    np.random.seed(seed)

    total_weighted_success = 0.0
    total_weight = 0.0
    success_count = 0
    weights = []

    for _ in range(N):

        delta = 0
        weight = 1.0

        while True:

            if np.random.rand() < alpha_tilde:
                delta += 1
```

```
                weight *= alpha / alpha_tilde
            else:
                delta -= 1
                weight *= (1 - alpha) / (1 - alpha_tilde)

            if delta >= L:
                total_weighted_success += weight
                success_count += 1
                break

            if delta < 0:
                break

        total_weight += weight
        weights.append(weight)

    p_estimate = total_weighted_success / N

    std_error = np.sqrt(
        p_estimate * (1 - p_estimate) / N
    )

    ci_low, ci_high = norm.interval(
        0.95,
        loc=p_estimate,
        scale=std_error
    )

    print(
        f"Success probability: "
        f"{p_estimate:.6f}"
    )

    print(
        f"95% confidence interval: "
        f"[{ci_low:.6f}, {ci_high:.6f}]"
    )

    print(
```

```
        f"Successful trajectories: "
        f"{success_count} of {N}"
    )

    if visualize:
        plt.figure(figsize=(8, 5))
        plt.hist(
            weights,
            bins=50,
            color='skyblue',
            edgecolor='black'
        )
        plt.title(
            'Distribution of trajectory weights'
        )
        plt.xlabel('Weight')
        plt.ylabel('Amount')
        plt.grid(True)
        plt.tight_layout()
        plt.show()

    return p_estimate, (ci_low, ci_high), success_count
```