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

«На правах рукопису»
УДК 004.056.55

«До захисту допущено»
В.о. завідувача кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2023 р.

Дипломна робота

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

зі спеціальності: 113 Прикладна математика
на тему: **«Дослідження атак централізації та саботажу на алгоритм консенсусу Casper»**

Виконав:
студент IV курсу, групи ФІ-94
Мельник Ілля Андрійович _____

Керівник:
професор кафедри ММЗІ, д.т.н., с.н.с.
Кудін Антон Михайлович _____

Рецензент:
Заступник директора департаменту безпеки Національного банку України, к.т.н., с.н.с.
Проскуровський Роман Васильович _____

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

Студент _____

Київ — 2023

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
В.о. завідувача кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2023 р.

ЗАВДАННЯ
на дипломну роботу

Студент: Мельник Ілля Андрійович

1. Тема роботи: «*Дослідження атак централізації та саботажу на алгоритм консенсусу Casper*», науковий керівник дипломної роботи: професор кафедри ММЗІ, д.т.н., с.н.с. Кудін Антон Михайлович,

затверджені наказом по університету №__ від «__» _____ 2023 р.

2. Термін подання студентом роботи: «__» _____ 2023 р.

3. Об'єкт дослідження: *протокол консенсусу Casper, реалізований у децентралізованій системі*

4. Предмет дослідження: *стійкість протоколу консенсусу Casper до атаки, коли принцип децентралізації в системі буде порушуватися*

5. Перелік завдань:

1) Аналіз протоколу Casper FFG;

2) Дослідження нових атак централізації та саботажу на протокол Casper FFG;

3) Оцінка стійкості протоколу Casper FFG до нових атак централізації та саботажу;

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: *презентація доповіді, 17 рисунків*

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

8. Дата видачі завдання: 10 вересня 2022 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	01-15 вересня 2022 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	Вересень-жовтень 2022 р.	Виконано
3	Вибір протоколу консенусу для дослідження	Жовтень-листопад 2022 р.	Виконано
4	Аналіз роботи протоколу Casper та оцінок стійкості до відомих атак централізації та саботажу	Листопад-січень 2022-2023 р.	Виконано
5	Аналіз нових атак централізації та саботажу та побудова оцінок їх успіху на протокол Casper	Лютий-квітень 2023 р.	Виконано
6	Проходження преддипломної практики	Квітень-травень 2023 р.	Виконано
7	Оформлення дипломної роботи	Травень-червень 2023 р.	Виконано

Студент _____ Ілля МЕЛЬНИК

Керівник _____ Антон КУДІН

РЕФЕРАТ

Кваліфікаційна робота містить: 48 стор., 17 рисунків, 1 таблицю, 10 джерел.

У даній роботі було досліджено стійкість протоколу консенсусу Casper FFG до атак централізації та саботажу. Для цього було обрано нові атаки: удосконалена атака Reorg (A Refined Reorg Attack), удосконалена атака живучості (A refined liveness attack) та атака Reorg з використанням ймовірнісної мережевої затримки (Reorg attack using probabilistic network delay). Метою дослідження є прорахунок та аналіз оцінок успіху розглянутих атак централізації та саботажу, щоб підтвердити чи спростувати вразливість протоколу Casper FFG до них. Об'єктом дослідження є протокол консенсусу Casper, реалізований у децентралізованій системі. Предметом дослідження є стійкість протоколу консенсусу Casper до атак, коли принцип децентралізації в системі буде порушуватися.

Результатами даного дослідження є оцінки успіху нових атак централізації та саботажу на протокол консенсусу Casper, їх порівняння та оцінка стійкості протоколу до описаних атак.

ПРОТОКОЛ КОНСЕНСУСУ, БЛОКЧЕЙН, CASPER, АТАКИ ЦЕНТРАЛІЗАЦІЇ

ABSTRACT

The qualification work contains: 48 p., 17 figures, 1 table, 10 sources

In this work, the resilience of the Casper FFG consensus protocol to centralization and sabotage attacks was investigated. New attacks were chosen for this: A Refined Reorg Attack, A Refined Liveness Attack and Reorg attack using probabilistic network delay. The purpose of the study is to calculate and analyze the success rates of the considered centralization and sabotage attacks in order to confirm or deny the vulnerability of the Casper FFG protocol to them. The object of research is the Casper consensus protocol implemented in a decentralized system. The subject of the study is the resistance of the Casper consensus protocol to attacks when the principle of decentralization in the system is violated.

The results of this study are evaluations of the success of new centralization and sabotage attacks on the Casper consensus protocol, their comparison, and an evaluation of the protocol's resistance to the described attacks.

CONSENSUS PROTOCOL, BLOCKCHAIN, CASPER, CENTRALIZATION ATTACKS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

PoW — протокол Proof-of-Work

PoS — протокол Proof-of-Stake

DPoS — протокол Delegated-Proof-of-Stake

Casper FFG — протокол Casper, версія Friendly Finality Gadget

BFT — Візантійська відмовостійкість

ВСТУП

Актуальність дослідження. Швидкий розвиток технологій призводить до того, що нещодавно побудовані моделі починають відставати від сьогоднішніх вимог. Надійність, швидкість та екологічність – основні критерії, які встановлюють поріг. Розглядаючи розвиток блокчейн-технологій, можна зрозуміти, що наразі протокол консенсусу Proof-of-Work має величезний недолік, а саме у використанні величезної кількості енергії майнерами та необхідністю постійно бути у мережі. Така проблема є досить критичною враховуючи обмеженість ресурсу енергії та забруднення, яке виникає в наслідок вироблення. На його заміну прийшов протокол Proof-of-Stake, що частково зміг вирішити питання у можливості зменшення споживання великої кількості енергії та розподіленню роботи. Але постає питання у надійності даної системи. Як поводитиметься система, якщо основний її принцип, децентралізація, буде порушений і як цьому можна запобігти?

Метою дослідження є оцінка стійкості відомого протоколу консенсусу Casper до атак централізації та саботажу. Для досягнення мети необхідно розв'язати **задачу дослідження**, яка полягає у побудові оцінок успіху атак централізації та саботажу на протокол Casper FFG. Для розв'язання задачі необхідно вирішити такі завдання:

1) Розглянути відомі протоколи консенсусу, а також їх переваги та недоліки;

2) Розглянути принцип роботи протоколу Casper та розглянути його стійкість до атак, що були вже оцінені;

3) Описати атаки централізації та саботажу, що не були оцінені для даного протоколу.

4) Вивести оцінки успішності цих атак на даний протокол та порівняти їх;

Об'єктом дослідження є протокол консенсусу Casper FFG,

реалізований у децентралізованій системі.

Предметом дослідження є стійкість протоколу консенсусу Casper FFG до атаки, коли принцип децентралізації в системі буде порушуватися.

При розв'язанні поставлених завдань використовувались такі *методи дослідження*: методи теорії імовірностей, математичної статистики, комбінаторного аналізу, теорії складності алгоритмів.

Наукова новизна отриманих результатів полягає у побудові оцінок успішності атак на протокол консенсусу Casper FFG для того, щоб підтвердити чи спростувати вразливість протоколу Casper FFG до них.

Практичне значення результатів полягає у розумінні чи необхідно вдосконалювати протокол консенсусу Casper FFG, щоб запобігти цим атакам або підтвердити надійність протоколу для використання сьогодні.

1 ТЕОРИТИЧНІ ВІДОМОСТІ

У даному розділі ми ознайомимося з основами, що будуть необхідні для подальшої нашої роботи. У першій секції коротко висвітлимо основні поняття та структурами з якими будемо працювати, а саме: алгоритми консенсусу в блокчейн-системах, протоколом Casper, алгоритмом візантійської відмовостійкості та задачею про «Візантійських генералів», а також з дослідженими атаками централізації та саботажу.

1.1 Головні терміни

Блокчейн – тип структури даних, що використовується для запису даних у розподілених обчислювальних мережах. Це послідовний у хронологічному порядку ланцюг блоків даних, отриманих учасниками протягом заздалегідь визначеного періоду часу. Зв'язок між цими блоками забезпечується використанням важкооборотної криптографічної хеш-функції (Сатоші Накамото, 2008) [4]. У міру того, як ланцюжок блоків буде збільшуватися, труднощі, пов'язані з перерахунком хеш-значення, також зростають, що робить будь-які зміни в минулих даних дорогими. Це зростання труднощів призводить до детермінованої гарантії незмінності даних.

Протоколи консенсусу використовуються вузлами в розподіленій мережі для прийняття послідовних рішень із можливо непослідовних альтернативних рішень. У консенсусному протоколі блокчейну рішення щодо одного блоку називається «несумісним» з іншим, якщо вони не знаходяться в одній гілці, і «сумісним», коли вони знаходяться в одній гілці. Якщо говорити точніше, то для протоколів консенсусу повинні виконуватися наступні властивості[7]:

Безпека: вузли не приймають несумісне рішення.

Детермінованість рішення: вузли неминуче (рано чи пізно) приймуть рішення.

Також варто зазначити, що протоколи консенсусу діляться на дві категорії: «синхронно безпечні», в яких безпека рішення гарантує час або якоюсь іншою умовою (наприклад, повідомлення повинні надійти до якогось часу) та «асинхронно безпечні», в яких немає обмеження в часі чи будь-якою іншою умовою за виключенням того, що всі надіслані повідомлення отримуються у будь-якому випадку.

1.2 Протокол Proof-of-Work

Протокол Proof-of-work (підтвердження праці) – це механізм, що використовується у децентралізованих мережах для узгодження консенсусу, тобто вузли можуть дійти згоди з приводу операцій, проведених у даній мережі. У блокчейні, даний алгоритм робить обчислювально складною операцію майнінгу нового блоку, використовуючи достатньо просте правило вибору розгалуження: «блок із найважчим ланцюгом є головою ланцюга»[10]. По суті, майнінг і є та сама робота, що ми маємо підтвердити. Це процес додавання нових блоків до ланцюга, що допомагає мережі слідувати правильному розгалуженню блокчейну. Чим важче гілка, тим більше блоків вона містить, а отже більше роботи було зроблено. Таким чином, результатом консенсусу буде просто ланцюг із найбільшим обсягом роботи, який в ідеалі продовжує зростати. Варто зазначити, що таке підтвердження роботи є достатньо дорогим через великі обсяги обчислювальної роботи та енергетичних витрат.

1.3 Протокол Proof-of-Stake

Proof-of-Stake (доказ частки) – ще один механізм, що використовується блокчейнами для досягнення розподіленого консенсусу. На відміну від Proof-of-work, де майнери доводять, що ризикують капіталом, витрачаючи енергію, в Proof-of-Stake валідатори вкладають капітал у формі криптовалюти цієї мережі, як заставу, яка у разі порушення правил, буде знищена[3]. Валідатор – це особа, що відповідає за перевірку того, що нові блоки, які розповсюджуються через мережу, дійсні. А у певний період часу, він може створювати та розповсюджувати нові блоки. Блоки у даному алгоритмі створюються у фіксованому темпі. Існує дві часові одиниці: слоти та епохи. Між валідаторами випадковим чином розподіляються слоти. Коли мережа знаходиться у певному слоті, валідатор, якому належить цей слот, відповідає за створення нового блоку та надсилання його на інші вузли в мережі. Коли ми пройдемо усі слоти, це буде сигнал того, що закінчується епоха. У разі виникнення розгалуження, валідатори голосують своїми частками за дійсну гілку. Якщо гілка набирає більш як дві третини голосів, то вона вважається дійсною. Таким чином, розпочинається нова епоха, а стара зберігається. Такий підхід забезпечує цілісність історії блоків.

1.4 Протокол Delegated Proof-of-Stake

Delegated Proof-of-Stake – це консенсусний механізм, що є різновидом класичного Proof-of-Stake. Створив алгоритм Ден Ларімер у 2013 році, а після використав його у своєму проєкті BitShares. Як і у протоколі PoW, у звичайному PoS виникає деяка нерівність серед користувачів: валідатор, що має більшу частку, має більшу винагороду, а отже ймовірність, що саме він буде створювати блок є більшою. З чого слідує, що особі з малим стейком невигідно майнити у даній системі. [6]

У DPoS користувачі обирають голосуванням делегатів, що будуть підтверджувати нові блоки. Причому вони можуть змінюватися, оскільки користувачі можуть голосуванням обрати інших делегатів. Вибір делегатів тепер залежить не тільки від їх частки, а і їх репутації, що буде підтверджувати їх надійність. Варто зазначити, що не усі делегати, можуть додавати нові блоки. Генерацією та розповсюдженням блоків займаються так звані *свідки. Свідок* це особа, що займається стейкінгом та має можливість стати делегатом. На відміну від звичайних делегатів, вони не мають доступу до генезисного блоку і не можуть змінювати параметри системи.

Плюси даного протоколу:

1) *Репутація.* Делегати обираються за допомогою демократичного процесу, що дозволяє їм створити собі репутацію, що і є ключовим мотивом для користувачів голосувати за них, а не просто за людей із найбільшими частаками (стейками).

2) *Швидкість.* DPoS досягає консенсусу досить швидко, оскільки мережа має обмеження щодо кількості делегатів. Зазвичай число варіюється від 20 до 100 делегатів, залежно від блокчейну. Таким чином, менша кількість делегатів дозволяє мережі досягти швидше консенсусу ніж у PoS чи PoW.

3) *Масштабованість.* Для доступу до більшості мереж DPoS, достатньо просто мати гаманець зі стейком.

Мінуси даного протоколу:

1) *Зловмисні власники токенів.* Оскільки в нас обмежена кількість делегатів, ймовірність, що серед них більшість виявиться зловмисниками є достатньо великою. Отже, система DPoS піддається ризику атаки 51%, тобто атаки, у якій залучено принаймні 51% делегатів, які діють зловмисно щодо мережі.

2) *Низький рівень децентралізації.* Оскільки системи DPoS зазвичай мають лише певну кількість делегатів, виникає питання, чи справді вони децентралізовані? Наприклад, мережу, яка має менш ніж 30

делегатів одночасно, можна вважати несправжньою децентралізованою.

3) *Постійна взаємодія.* Система DPoS вимагає від учасників голосування за делегатів, щоб забезпечити роботу мережі. Через це людям, які користуються мережею, необхідно залишатися активними; інакше мережа не працюватиме.

1.5 Протокол Casper

Casper – механізм досягнення консенсусу у блокчейні, що поєднав у своїй реалізації два відомих механізми: PoW та PoS. Хоча, наразі ведеться розробка заміни PoW на іншу модифікацію, що могло б покращити роботу протоколу. Ми ж будемо розглядати просту версію Casper, що називається «Casper the Friendly Finality Gadget» [2]. Розробники даної версії є Віталій Бутерін та Вірджіл Гріффіс.

Основна функція даного протоколу відповідає за вибір унікального ланцюга, який представляє канонічний облік транзакцій. При цьому він забезпечує безпеку системи, а от життєздатність системи вже залежить від виду функції пропозиції блоків. Припустимо, що зловмисники повністю контролюють функцією пропозиції блоків. Для кращого розуміння введемо поняття контрольні точки – кінцеві блоки гілок нашої системи. Casper захищає систему, роблячи вибір з двох конфліктних контрольних точок, але зловмисники зможуть перешкодити Casper завершити будь-які контрольні точки у майбутньому.

Перейдемо до опису самої схеми. Припускаємо, що існує фіксований набір валідаторів і механізм пропозиції блоків (наприклад, механізм пропозиції на основі PoW), що створює дочірні блоки наявних блоків, утворюючи дерево блоків, яке постійно зростає. Коренем цього дерева зазвичай називають «блоком генезису». За звичайних обставин ми очікуємо, що механізм пропозиції пропонуватиме блоки один за одним у зв'язаному списку(тобто кожен «батьківський» блок має рівно один «дочірній» блок). Але у випадку затримки мережі або навмисних атак

механізм пропозиції неминуче іноді створюватиме кілька нащадків одного предка. Завдання Casper полягає в тому, щоб вибрати одного нащадка від кожного предка, таким чином вибравши один канонічний ланцюг із дерева блоків.

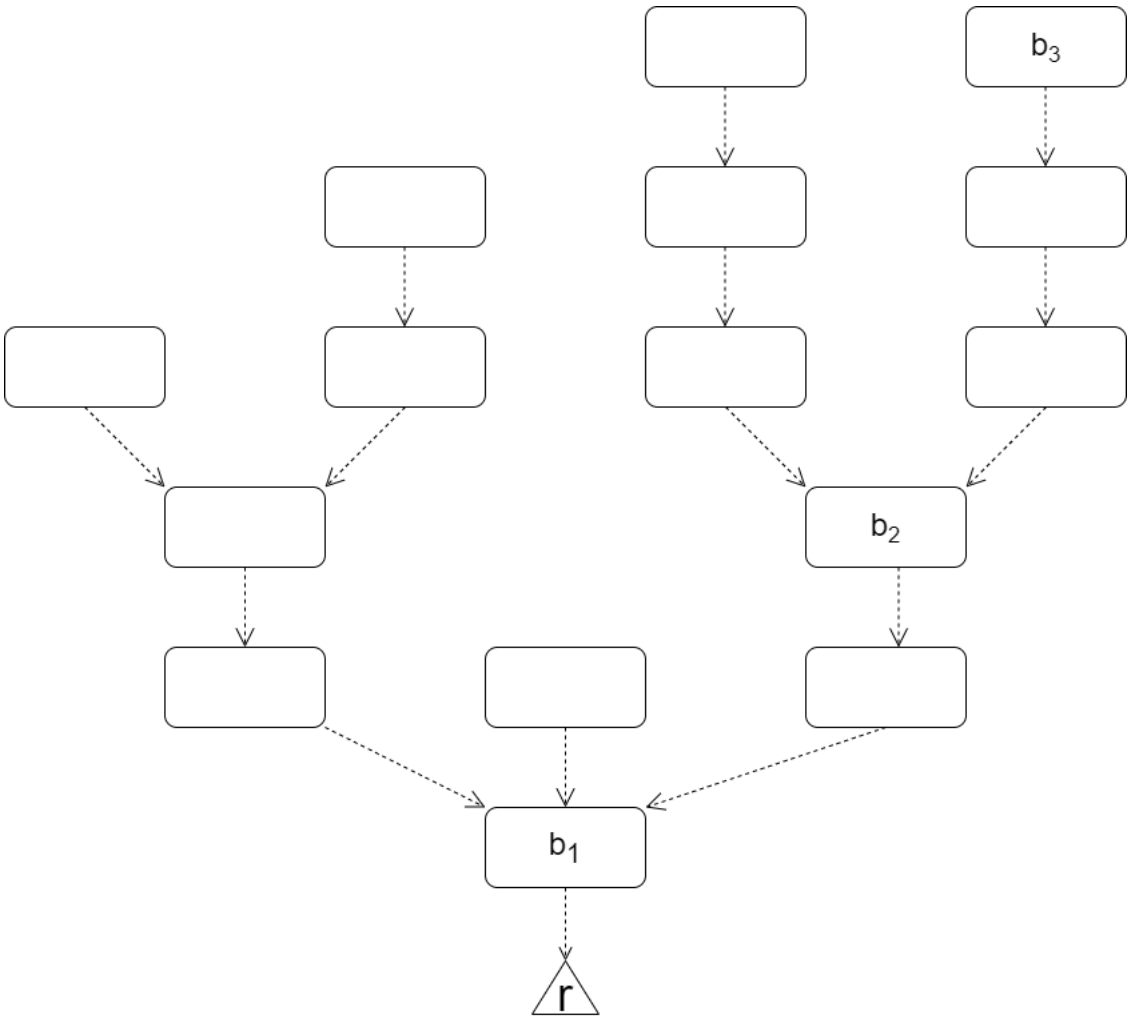

Рисунок 1.1 – Дерево контрольних точок. Пунктирна лінія позначає 99 блоків між контрольними точками, які представлені округленими прямокутниками. Корінь дерева позначається буквою r.

Замість того, щоб мати справу з повним деревом блоків, з метою ефективності Casper розглядає лише піддерево контрольних точок, які утворюють дерево контрольних точок (рис. 1.1). Блок генезису є контрольною точкою, і кожен блок, висота якого в дереві блоків (або номер блоку) є кратним 100, також є контрольною точкою. «Висота

контрольної точки» блоку з висотою блоку $100 * k$ дорівнює просто k; еквівалентно, висота $h(c)$ контрольної точки c є кількістю елементів у ланцюгу контрольної точки, що тягнеться від c (не включно) до кореня вздовж батьківських ланок (рис. 1.2).

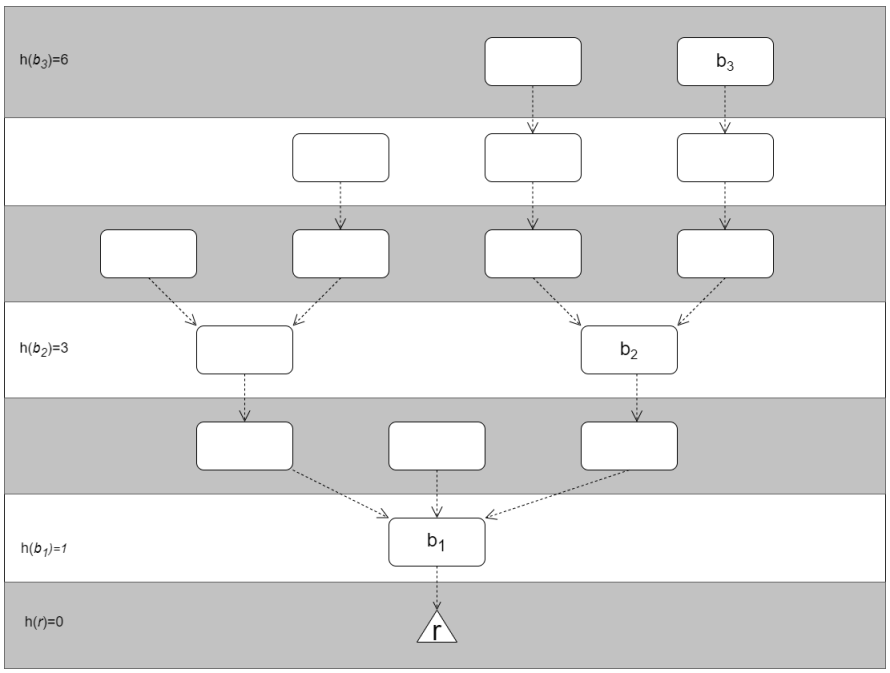

Рисунок 1.2 – Ілюстрація дерева контрольних точок та функції висоти.

Кожен валідатор має депозит, коли валідатор приєднується, його депозит дорівнює кількості внесених монет. Після приєднання депозит кожного валідатора зростає та зменшується разом із винагородами та штрафами. Валідатори можуть транслювати голосове повідомлення, що містить чотири частини інформації (таблиця 1.1): дві контрольні точки s і t разом із їхніми висотами $h(s)$ і $h(t)$. Ми вимагаємо, щоб s був предком t у дереві контрольних точок, інакше голосування вважається недійсним. Якщо відкритого ключа валідатора ν немає в наборі валідатора, голосування вважається недійсний. Разом із підписом валідатора запишемо ці голоси у формі $(\nu, s, t, h(s), h(t))$.

Дамо визначення наступним термінам:

1) Посилання на множину більшості майнерів – це впорядкована

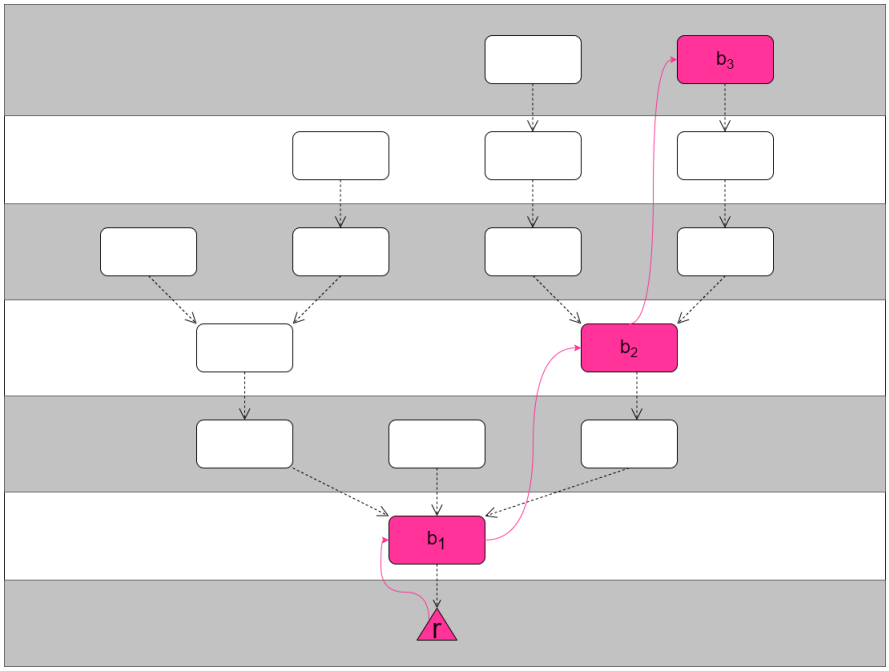

Рисунок 1.3 – Ілюстрація вирівняного ланцюга всередині дерева контрольних точок.

пара контрольних точок (a, b), позначається $a \to b$, така що принаймні $\frac{2}{3}$ валідатори (за депозитом) опублікували голоси за джерелом a та ціллю b. Посилання на множину більшості майнерів можуть пропускати контрольні точки, тобто це цілком нормально для $h(b) > h(a) + 1$. На рисунку 1.3 показано три чіткі множини більшості майнерів – ланки рожевого кольору: $r \to b1, b1 \to b2$ і $b2 \to b3$.

2) Дві контрольні точки a і b називаються конфліктними тоді й тільки тоді, коли вони є вузлами в різних гілках, тобто жодна не є предком або нащадком іншої.

3) Контрольна точка c називається виправданою, якщо вона є кореневою, або існує посилання на множину більшості майнерів $c' \to c$, де контрольна точка c' є виправданою. На рисунку 1.3 показано ланцюг із чотирьох виправданих блоків.

4) Контрольна точка c називається завершеною, якщо вона є коренем або вона виправдана та існує посилання на множину більшості майнерів $c' \to c$, де c' є прямим дочірнім елементом c. Еквівалентно, контрольна

Таблиця 1.1 – Схема одного повідомлення VOTE, позначеного $(\vee, s, t, h(s), h(t))$.

Змінні	Значення
s	хеш будь-якої виправданої контрольної точки («джерело»)
t	будь-який хеш контрольної точки, який є нащадком s («ціль»)
$h(s)$	висота контрольної точки s у дереві контрольних точок
$h(t)$	висота контрольної точки t у дереві контрольних точок
S	підпис $(s, t, h(s), h(t))$ із закритого ключа валідатора $\vee$

точка c завершена тоді й тільки тоді, коли контрольна точка c виправдана, існує посилання на множину більшості майнерів $c' \to c$, контрольні точки c і c' не конфліктують, і $h(c') = h(c) + 1$.

Тепер визначимось із валідаторами, оскільки необхідно мати можливість змінювати їх. Нові валідатори повинні мати можливість приєднатися, а чинні валідатори повинні мати можливість вийти. Щоб досягти цього, визначимо династію блоку. Династія блоку b – це кількість завершених контрольних точок у ланцюгу від кореня до кінцевого блоку b. Коли повідомлення про депозит потенційного валідатора включено до блоку з династією d, тоді валідатор ν приєднається до валідатора, встановленого в першому блоці з династією $d + 2$. Ми називаємо $d + 2$ початковою династією цього валідатора, $\mathrm{DS}(\nu)$. Щоб залишити встановлений валідатор, він повинен надіслати повідомлення «відкликати». Якщо повідомлення про відкликання валідатора ν включено до блоку з династією d, воно так само залишає валідатор, встановлений у першому блоці з династією $d + 2$; ми називаємо $d + 2$ кінцевою династією валідатора, $\mathrm{DE}(\nu)$. Якщо повідомлення про відкликання ще не включено, тоді $\mathrm{DE}(\nu) = \infty$. Коли валідатор ν залишає набір валідаторів, відкритому ключу валідатора назавжди забороняється повторно приєднуватися до набору валідаторів. Це усуває необхідність обробляти кілька початкових/кінцевих династій для одного ідентифікатора. На початку останньої династії депозит валідатора

блокується на тривалий період часу, який називається затримкою зняття, перш ніж депозит буде знято. Якщо під час затримки виведення валідатор порушує будь-яке правило, депозит скорочується. Ми визначаємо дві функції, які генерують дві підмножини валідаторів для будь-якої заданої династії d, набір наступного валідаторів і набір кінцевого валідатора. Вони визначаються як

$$\begin{aligned} \mathcal{V}_f(d) &\equiv \{\nu : DS(\nu) \leqslant d < DS(\nu)\}; \\ \mathcal{V}_r(d) &\equiv \{\nu : DS(\nu) < d \leqslant DS(\nu)\}. \end{aligned}$$

Зауважимо, що наступний набір валідаторів династії d є кінцевим набором валідаторів династії $d + 1$, тобто $\mathcal{V}_f(d) = \mathcal{V}_r(d + 1)$. Щоб підтримувати ці динамічні набори валідаторів, перевизначимо посилання на множину більшості майнерів та завершену контрольну точку наступним чином:

1) Упорядкована пара контрольних точок (s, t), де t знаходиться в династії d, має посилання на множину більшості майнерів, якщо принаймні $\frac{2}{3}$ із набору наступних валідаторів династії d мають опубліковані голоси $s \to t$ і принаймні $\frac{2}{3}$ із кінцевого валідатора набір династії d мають опубліковані голоси $s \to t$.

2) Раніше контрольну точку c називали завершеною, якщо c виправдана й існує посилання на множину більшості майнерів від $c \to c'$, де c' є дочірнім для c. Тепер ми додаємо умову, згідно з якою c є завершеною, якщо тільки голоси за посилання на множину більшості майнерів $c \to c'$, так само як посилання на множину більшості майнерів, що виправдовує c, включені в ланцюг блоків c' і перед дочірнім елементом c', тобто перед блоком номер $h(c') * 100 + 1$.

1.6 Задача про «Візантійських генералів»

Як ми знаємо, будь-яка комп'ютерна система складається з багатьох компонентів, що пов'язані один з одним. І якщо система зможе впоратися з випадком, коли один чи декілька її компонентів вийдуть з ладу, то можемо вважати таку систему надійною. Несправний компонент може демонструвати поведінку, яку ми можемо не бачити, а саме надсилання хибних чи суперечливих даних іншим компонентам. Подолання даної проблеми абстрактно можна описати як задачу про «Візантійських генералів»[8]. Нехай в нас є деяка кількість таборів, що розміщені у різних місцях довкола ворожого міста. Кожним табором керує один генерал. Для спілкування між собою вони використовують месенджер. Тепер їм необхідно домовитися про спільний напад. Кожен з генералів може вибрати один з двох варіантів голосу: атака чи відступ. Однак деякі генерали можуть бути зрадниками й будуть намагатися перешкодити досягненню згоди та прийняттю успішного плану. Для того, щоб зрадники не порушили плани, необхідно, щоб при прийнятті рішення, голоси надійних генералів переважали голоси зрадників. Було доведено, що достатньо щоб кількість голосів надійних генералів мала вагу $\frac{2}{3}$ від загальної кількості голосів.

1.7 Візантійська відмовостійкість (BFT)

Візантійська відмовостійкість – це властивість системи, яка здатна протистояти класу відмов, що випливають із проблеми візантійських генералів. Це означає, що система BFT може продовжувати роботу, навіть якщо деякі з вузлів виходять з ладу або діють зловмисно[1].

Існує більше одного можливого розв'язання проблеми візантійських генералів, а отже, і декілька способів побудови системи BFT. Так само існують різні підходи до блокчейну для досягнення візантійської

відмовостійкості, і це призводить нас до так званих консенсусних алгоритмів.

Детально ознайомившись із самим протоколом, а також з умовою візантійської відмовостійкості, можемо перейти до опису атак на протокол Casper. Ми розглянемо дві найвідоміші атаки на системи з механізмом PoS: *довгострокові перегляди (long range revisions)* та *катастрофічні збої (catastrophic crashes).*

1.8 Атака довгострокові перегляди (long range revisions)

Затримка вилучення після закінчення династії валідатора вводить припущення про синхронність між валідаторами та клієнтами. Після того, як коаліція валідаторів відкликає свої депозити, якщо ця коаліція мала більше $\frac{2}{3}$ депозитів у минулому, то вони можуть використати свою історичну множину більшості майнерів, щоб завершити конфліктні контрольні точки, не боячись бути покараними(оскільки вони вже зняли свої гроші). Це називається атакою довгострокового перегляду (рисунок 1.4).

Щоб запобігти атаці довгострокових переглядів, необхідно використати правило вибору розгалуження, щоб ніколи не повертати завершений блок, а також очікування, що кожен клієнт «зареєструється» та отримає повне оновлене уявлення про ланцюг з певною регулярною частотою(приблизно раз на 1-2 місяці). Розгалуження «довгострокового перегляду», яке завершує старіші блоки, просто ігноруватиметься, оскільки всі клієнти вже бачать завершений блок на такій висоті та відмовляться повертати його.

Для того, щоб довести, чому це працює, введемо наступні припущення:

1) Між двома клієнтами існує максимальна затримка зв'язку δ, тому, якщо один клієнт отримає якесь повідомлення в момент часу t, усі інші

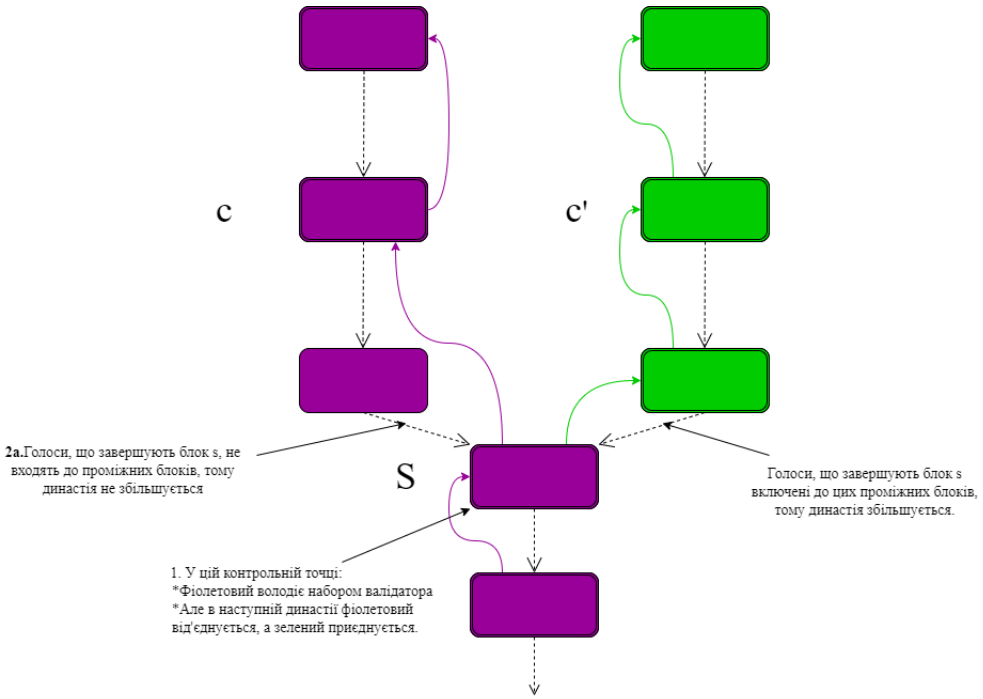

Рисунок 1.4 – Атака з наборами динамічних валідаторів. Без механізму з'єднання набору валідатора можливо, що дві конфліктні контрольні точки c і c' можуть бути завершені без жодного валідатора. У цьому випадку c і c' мають однакову висоту, що порушує правило безпеки системи, але оскільки набори валідаторів, що завершують c і c', не перетинаються, ніхто не буде покараний.

клієнти гарантовано отримають його до часу $t + \delta$. Це означає, що ми можемо говорити про «часове вікно» $[t_{min}, t_{max}]$, протягом якого блок був отриманий мережею, з шириною $t_{max} - t_{min}$ не більше δ.

2) Припускаємо, що всі клієнти мають локальні годинники, які ідеально синхронізовані (будь-яка невідповідність може розглядатися як частина затримки зв'язку δ).

3) Блоки повинні мати мітки часу. Якщо клієнт має місцевий час T_L, тоді він зможе відхилити блоки, часові позначки яких T_B задовольняє умові – $T_B > T_L$ (тобто в майбутньому). І вони відмовляться прийняти доопрацьовані (але можуть прийняти як частину ланцюжка) блоки, де $T_B < T_L - \delta$ (тобто далеко в минулому).

4) Якщо валідатор бачить порушення розрізання в момент часу t (це час, коли він отримує останній із двох голосів, необхідних для порушення розрізання), він відхиляє блоки з часовими мітками $> t+2\delta$, що є частиною ланцюжків, які ще не включали це різання доказів.

Припустимо, що великий набір порушень розрізу призводить до двох конфліктних завершених контрольних точок, c_1 і c_2. Якщо два часові проміжки не перетинаються, тоді всі валідатори погоджуються, яка контрольна точка була першою, і всі дотримуються правила не повертати завершені контрольні точки. В такому випадку ніяких проблем не виникає. Якщо два часові проміжки перетинаються, тоді ми можемо розглянути випадок наступним чином. Нехай часовий проміжок c_1 буде $[0, \delta]$, а часовий проміжок c_2 буде $[\delta - E, 2\delta - E]$. Тоді мітки часу для обох дорівнюють принаймні 0. До моменту часу 2δ гарантується, що всі клієнти помітили порушення розрізу, тому вони відхиляють блоки з міткою часу $> 4\delta$, ланцюжки яких ще не включали транзакцію-доказ. Отже, якщо $\omega > 4\delta$, гарантовано, що зловмисники втратять свої депозити в усіх ланцюгах, які приймає будь-який клієнт, де ω – це «затримка виведення» (рисунок 1.5), тобто затримка між кінцевою епохою та моментом, коли валідатори фактично отримати свої депозити назад.

Через мережеві затримки можливо, що клієнти не погодяться з тим, чи було подано певний доказ урізання в певний ланцюжок «вчасно» чи вони прийняли його надто пізно. Однак це лише збій життєздатності системи, а не збій безпеки, і ця ситуація не послаблює наші претензії щодо безпеки, оскільки вже відомо, що пошкоджений механізм пропозиції (який буде необхідним для запобігання включенню доказів) може перешкодити остаточності. Ми також можемо обійти проблему тайм-аутів включення доказів, неофіційно стверджуючи, що атаки будуть короткочасними, тому що валідатори можуть сприйняти тривалий ланцюжок без урізання доказів як атаку та перейдуть до іншої гілки, підтримуваної чесною меншістю валідаторів, які не є частиною атаки, таким чином зупиняючи атаку та вражаючи зловмисника.

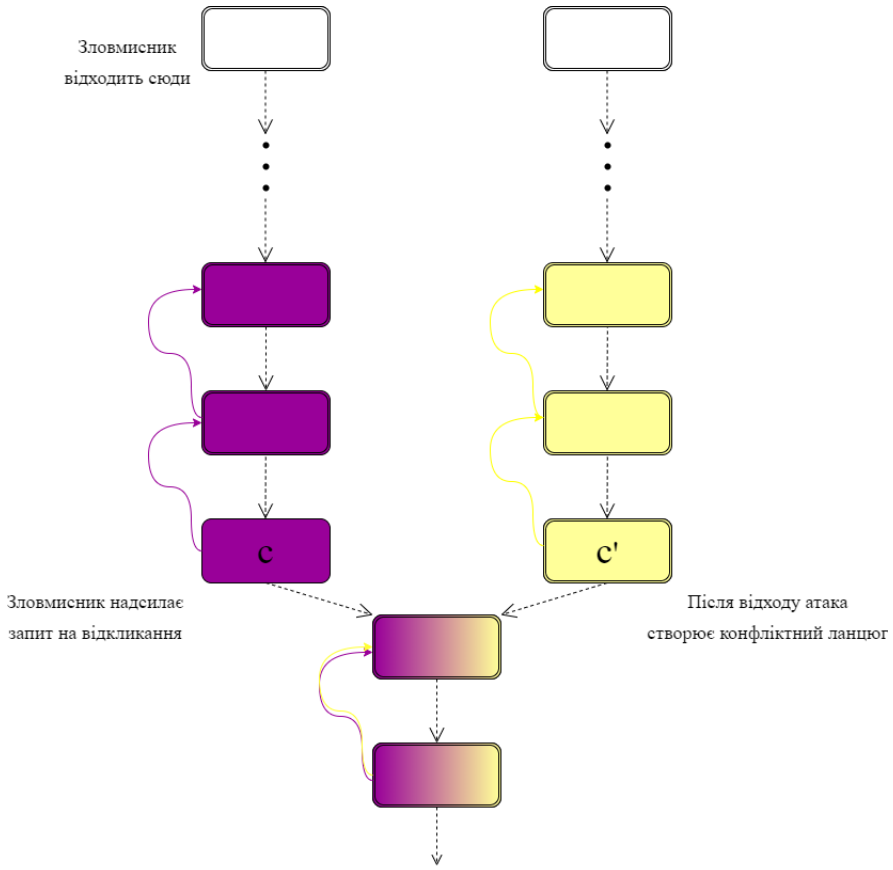

Рисунок 1.5 – Атака довгострокові перегляди. Поки клієнт отримує повні знання про виправданий ланцюжок через регулярні проміжки часу, він не буде вразливим до атаки на великій відстані.

1.9 Атака катастрофічних збоїв (catastrophic crashes)

Припустимо, що більше $\frac{2}{3}$ з валідаторів виходить з ладу одночасно, тобто вони більше не підключені до мережі через розділ мережі, збій комп'ютера або самі валідатори виявилися загрозливими. Інтуїтивно зрозуміло, що з цього моменту неможливо створити множину більшості майнерів, а отже, завершити контрольні точки в майбутньому. Ми можемо врятуватися від цього, започаткувавши недостатня активність, що повільно виснажує депозит будь-якого валідатора, який не голосує за

контрольні точки, доки врешті-решт розмір його депозиту не зменшиться настільки, що сума часток валідаторів, які голосують, буде становити множину більшості майнерів. Найпростіша формула виглядає приблизно так: «у кожну епоху валідатор із розміром депозиту D не голосує, він втрачає $D * p$ (для$0 < p < 1$)». Цей злитий ефір можна спалити або повернути валідатору через ω днів.

Недостатня активність створює можливість завершення двох конфліктних контрольних точок без урізання будь-якого валідатора (як на рисунку 1.6), при цьому валідатори втрачають гроші лише на одній із двох контрольних точок. Припустимо, що валідатори розділені на дві підмножини, причому підмножина $\mathcal{V}_A$ голосує за ланцюжок A, а підмножина $\mathcal{V}_B$ голосує за ланцюжок B. У ланцюжку A депозити $\mathcal{V}_B$ будуть зменшуватися внаслідок неактивності, і навпаки, що призведе до того, що кожна підмножина матиме множину більшості майнерів у відповідному ланцюжку, дозволяючи дві конфліктні контрольні точки, які будуть завершені без явного скорочення будь-яких валідаторів (але кожна підмножина втратить значну частину свого депозиту в одному з двох ланцюжків через неактивність). Якщо така ситуація трапляється, то кожен валідатор повинен просто віддати перевагу тій завершеній контрольній точці, яку він побачив першим.

Точний алгоритм відновлення після цих різних атак залишається відкритою проблемою. Є припущення, що валідатори можуть виявляти явно зловмисну поведінку (наприклад, не включаючи докази) і вручну створювати «м'який форк меншості». Цей міноритарний форк можна розглядати як власний блокчейн, який конкурує з мажоритарним ланцюгом на ринку, і якщо мажоритарний ланцюг справді керується зловмисними зловмисниками, які змовляються, тоді можна припустити, що ринок віддасть перевагу міноритарному форку.

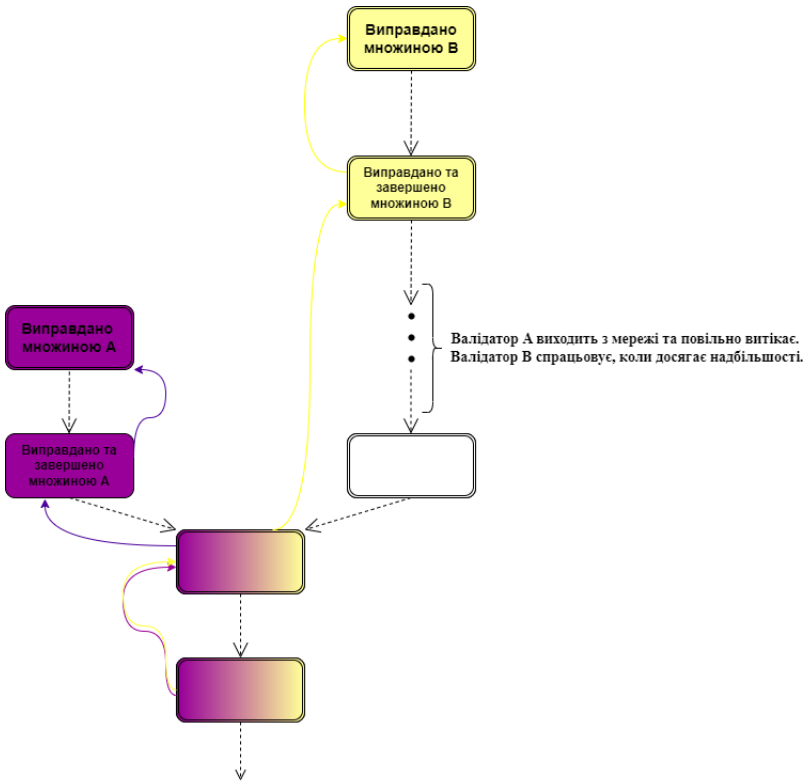

Рисунок 1.6 – Недостатня активність. Контрольну точку ліворуч можна завершити відразу. Контрольну точку праворуч можна завершити через деякий час, коли запаси офлайн-валідатора достатньо вичерпаються.

Висновки до розділу 1

У даному розділі, ми ознайомилися з протоколами консенсусу, їх різновидами та особливостями. Детально розглянули протокол Casper, задачу «Візантійських генералів», алгоритм візантійської відмовостійкості та розглянули дві найпоширеніші атаки на протоколи, що використовують механізм консенсусу Proof of Stake.

Аналізуючи алгоритми консенсусу, DPoS є оптимальним для використання, оскільки він успадковує усі властивості звичайного PoS та має чудову систему обрання валідаторів, оскільки це запобігає можливості, що зловмисники будуть становити більшість у системі та порушувати умову BFT, яка забезпечує децентралізацію у нашій системі. Саме такий принцип і використовує протокол Casper FFG.

Розглянуті атаки підтверджують факт того, що атаки централізації та саботажу можливі у даному протоколі, а отже необхідно перевірити чи достатньо стійкий протокол до нових атак, що наразі потенційно можуть становити загрозу для мережі.

2 УДОСКОНАЛЕНІ АТАКИ ЦЕНТРАЛІЗАЦІЇ ТА САБОТАЖУ

У даному розділі буде розглянуто алгоритми реалізації [5] трьох атак та побудовані оцінки ймовірності їх реалізації. Особливість даних атак полягає в тому, що зловмисник працює на зупинку нашого консенсусу та спотворення послідовного створення блоків.

2.1 Удосконалена атака Reorg (A Refined Reorg Attack)

Перша атака *A Refined Reorg Attack* є модифікацією звичайної *reorg* атаки. В даній модифікації можна зменшити кількість необхідних зловмисників, від набору лінійного розміру до набору постійного розміру. Насправді для одноблокового *reorg* достатньо лише одного зловмисника.

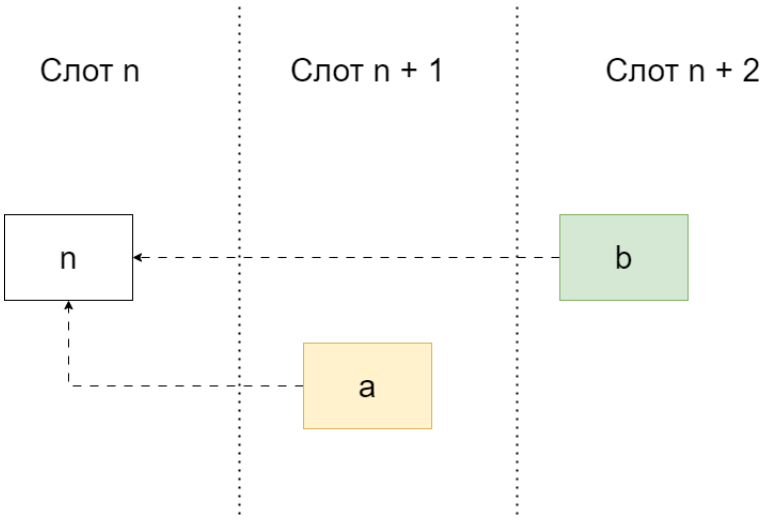

Рисунок 2.1 – Розгалуження, що виникло внаслідок атаки. Блок a створив зловмисник, блок b – чесний валідатор.

Надалі розпишемо покроково реалізацію даної атаки:

1) На початку слота n + 1 зловмисник приватно створює блок n + 1, що буде нащадком блоку n і приватно засвідчує його. Чесні валідатори не бачать блок n + 1 і тому підтверджують як останній блок ланцюга – блок n.

2) На початку наступного слота чесний валідатор пропонує блок n + 2. Далі можливі два варіанти подій:

а) Припускаючи, що затримка мережі нульова, зловмисник публікує приватний блок і атестацію зі слота n + 1 одночасно з блоком n + 2. Чесні валідатори тепер бачать як блок n + 1 (і його одну атестацію), так і блок n + 2. Ці блоки конфліктують, оскільки мають одного і того ж нащадка – блок n. За таких умов блок n + 1 успадковує всю вагу блоку n, зокрема, чесні атестації зі слота n + 1, що голосують за блок n.

б) Зловмисник залишає свій блок і атестації приватними. Таким чином, чесні валідатори вважатимуть кінцевим блоком n + 2.

3) Якщо валідатор опублікував свій блок та атестації разом зі створенням блоку n + 2, то чесні валідатори будуть голосувати за блок n + 1, оскільки він має більшу вагу через атестацію зловмисників зі слота n + 1. В іншому випадку, чесні валідатори будуть голосувати за блок n + 2, а зловмисник приватно атестує блок n + 1 використовуючи атестації слота n + 2, після чого публікує блок та атестації[9] (рисунок 2.2).

4) Нарешті, на початку слота n + 3 чесний валідатор пропонує блок n + 3, який вказує на блок n + 1 як на його предка. Це фактично знищує блок n + 2 і завершує атаку reorg.

Розглянута вище атака показує, що той, хто створює блок та керує одним членом групи в тому самому слоті, може успішно виконати *1–reorg*. Це ще обумовлено тим, що зловмисник може віддати дві свої атестації за блок a, використовуючи два слоти – n + 1 та n + 2 відповідно. Тобто, якщо коефіцієнт зловмисників у системі $\beta > 0.25$, тоді частка віддана за блок a буде переважати по вазі частку більшості чесних валідаторів. Для побудови оцінки уявимо, що це атака якоїсь довільної довжини k. Нехай

кількість чесних валідаторів у будь-якому комітеті буде дорівнювати $W_{honest} = W \cdot (1 - \beta) \leqslant W$. Тоді для успішної атаки довжиною $k > 1$, зловмисник повинен контролювати $W_{honest} \cdot (k - 1) + 1$ валідаторів, оскільки він компенсує голоси чесних членів комітету в перших $k - 1$ слотах і використовує наведену вище вдосконалену стратегію атаки в останньому слоті. Також варто зазначити, що коротші *reorg* атаки є більш здійсненні.

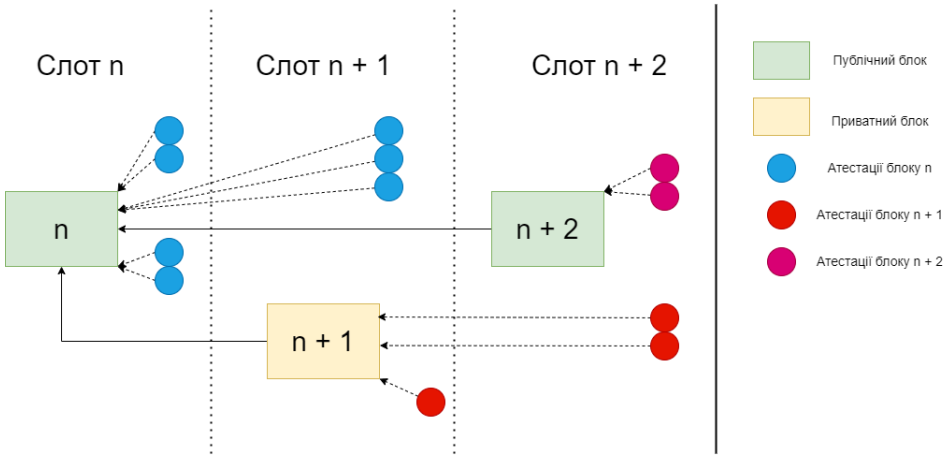

Рисунок 2.2 – Приклад використання 1-блокового розгалуження для виконання Reorg довжиною 1. Зловмисник може підтвердити свій приватний блок n + 1, використовуючи атестації обох слотів n + 1 і n + 2. Таким чином, атестації зловмисника, кожна з яких становить меншість у системі, на блоці n + 1 можуть перевищувати кількість чесних атестацій на блоці n + 2. До того як зловмисник звільнить блок n + 1, блок n + 2 є нащадком ланцюга. Після звільнення блоку n + 1 чесні вузли бачать, що вага блоку n + 1 дорівнює 3 (кількість червоних кульок), а вага блоку n + 2 дорівнює 2 (кількість рожевих кульок). У результаті алгоритм консенсусу визначає n + 1 як голову ланцюга, оскільки це блок з найбільшою вагою, який є нащадком блоку n, а блок n + 2 втрачає зв'язок і знищується.

Тому для подальших обрахунків, будемо вважати, що атака

виконується з тактом в один слот. Розпишемо ситуацію, зображену на рисунку 2.1. Припустимо, що в нас досить стійка система, де чесні валідатори становлять більшість (тобто $W_{honest} \geqslant \frac{2}{3}$). Покладемо за основу, що вибрати та проголосувати за блок можна з ймовірністю $p = \frac{1}{2}$. Для успіху необхідно, щоб за блок зловмисника a проголосувала більшість, тобто не менше ніж $\frac{W}{2}$. Врахуємо те, що усі валідатори, що належать до групи зловмисників, точно проголосують за блок a. Тому необхідно обрахувати скільки чесних валідаторів необхідно для перемоги блоку a. Якщо коефіцієнт зловмисників в мережі дорівнює β, то загальна кількість зловмисників – $W\beta$. Отже, необхідна кількість голосів чесних валідаторів буде дорівнювати:

$$\frac{W}{2} - W\beta = W(\frac{1}{2} - \beta)$$

Нехай подія A – успіх l-тої кількості атак у мережі за епоху. Обрахуємо ймовірність цієї події:

$$P = P(A) = P(X = l) = \lambda^l \frac{e^{-\lambda}}{l!} = \beta^l \frac{e^{-\beta}}{l!}$$

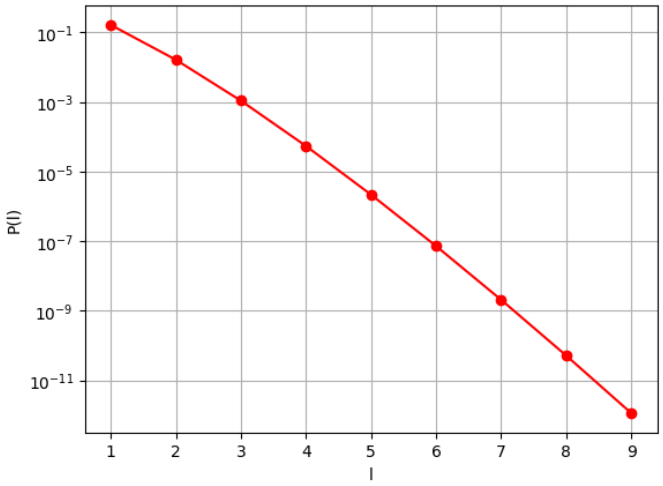

Рисунок 2.3 – Графік залежності ймовірності успіху атаки від кількості виконаних атак за епоху. Коефіцієнт зловмисників в системі $\beta = 0.2$.

Аналізуючи формулу ймовірності P можна зробити висновок, що з кожним наступним кроком ймовірність успіху атаки буде експоненційно спадати й швидко прямувати до 0, що можна побачити на рисунку 2.3. Це обумовлено тим, що на відміну від найпершої спроби атаки, де зловмисник може використати атестації з двох слотів, надалі така операція буде неможлива у зв'язку з тим, що атестації з попереднього слота вже використані. Також важливим фактором є те, що з часом чесні валідатори починають запідозрювати зловмисника у спотворенні ланцюга блоків. Це дуже важливий фактор, оскільки ми припускаємо, що у деяких випадках зловмисникам все ж знадобляться атестації чесних валідаторів для переваги над чесним блоком b.

2.2 Удосконалена атака живучості (A refined liveness attack)

Друга атака є доволі перспективною у плані реалізації на відміну від попередньої атаки. Припускається, що при успішній реалізації, навіть зловмисник, що контролює лише 15% акцій зміг би зупинити PoS Ethereum без затримки суперницької мережі (тут діє принцип: чим більше валідаторів, тим менше частка). Припускається, що розповсюдження випадкових повідомлень є достатньо передбачуваними, щоб надати зловмиснику необхідну інформацію для контролю над тим, скільки валідаторів бачать, які повідомлення і коли.

Нагадаємо, що наявна балансувальна атака складається з двох етапів: спочатку зловмисники, що створюють блоки ініціюють два конкурентних ланцюги – називають їх *лівим* і *правим*. Тоді достатньо кількох зловмисних голосів на слот, виданих за ретельно вибраних обставин, щоб скерувати голоси чесних валідаторів для утримування системи у зв'язку між двома ланцюгами та, як наслідок, зупинити консенсус.

Розглянемо детальніше алгоритм атаки (рисунок 2.4). Зауважимо, що для виконання нам необхідно два валідатори, які будуть у даній епосі

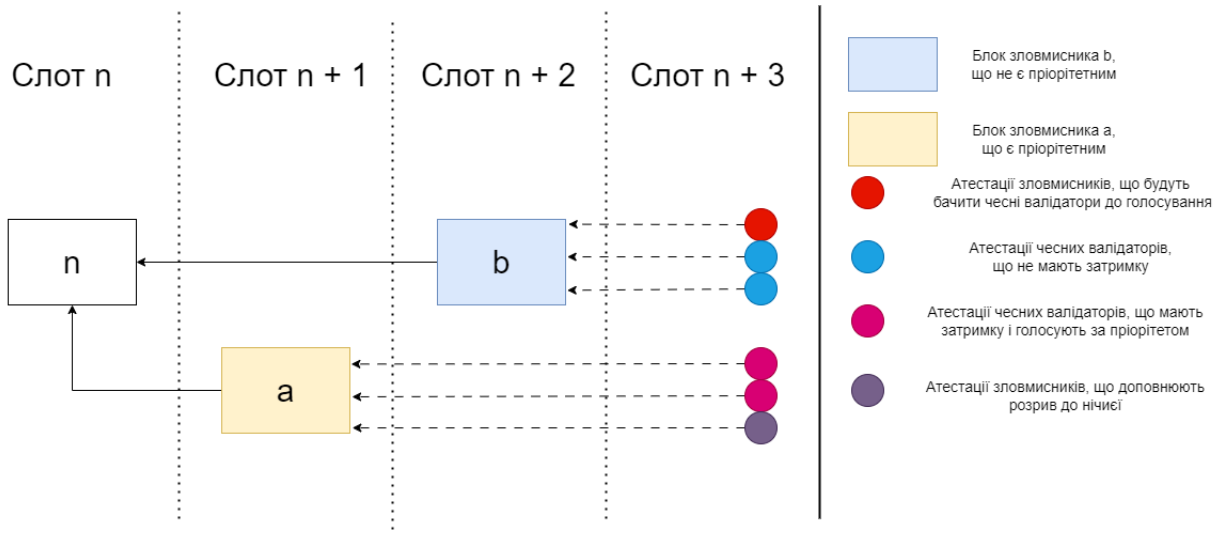

Рисунок 2.4 – Удосконалена атака живучості. Чесні валідатори розбиваються на половину у голосуванні, а зловмисник доважує одну з гілок до нічиєї.

створювати блоки, а за нічиєї у слотах i та $i+1$ пріоритет для лідерства буде мати блок у слоті i:

1) Дочекатися сприятливих умов, за яких у слотах i та $i+1$ створювати блоки будуть зловмисники.

2) Зловмисники відмовляються від блоків, що створили. Таким чином інші валідатори не зможуть проголосувати за будь-який із цих блоків, а система вважає це як нічию. Умовно позначимо ці блоки як a (слоту i) та b (слоту $i+1$).

3) Налаштувати час затримки T_{delay} так, що $\sim W_{honest}/2$ чесних валідаторів бачитиме результат голосування перед тим як віддати голос у слоті $i+2$, а інша – лише після.

4) Зловмисники віддають голос за блок b, використовуючи члена групи зі слота $i+1$. В результаті одна половина чесних валідаторів бачитимуть лідером блок b і голосуватимуть за нього, а інша половина буде голосувати керуючись фактом того, що пріоритет за блоком a.

5) Зрівняти голосування до нічиєї голосами членів групи зловмисників, що залишилися.

Якщо звести голосування до аналогії з підкиданням монети, то за *ліворуч* (блок a) і *праворуч* (блок b) проголосують $W_{honest}/2$ валідаторів відповідно з розривом $\mathcal{O}(\sqrt{W_{honest}})$. Отже, достатньо $\mathcal{O}(\sqrt{1/W_{honest}})$ частки ставки, щоб балансувати голосування до рівного та тримати систему в незавершеному стані.

Тепер перейдемо до обрахування часу затримки T_{delay}, що необхідний нам для реалізації нашої атаки. Почнемо з того, що зловмисник повинен дочекатися сприятливих умов для реалізації своєї атаки. Епоха вважається сприятливою, якщо блоки у слотах 0 та 1 створюють зловмисники. Користуючись фактом, що в нашому протоколі вибір комітету відбувається кожної нової епохи, така комбінація можлива з імовірністю β^2. Тобто в середньому треба чекати $1/\beta^2$ епох, щоб розпочати атаку. Припустимо, що наші умови сприятливі. Зловмисники, що створили блоки у слотах 0 та 1 пропонують два ланцюги: *лівий* і *правий*. Варто уточнити, що це не є порушенням правил протоколу. Обидва відмовляються від своїх пропозицій, щоб жоден із чесних валідаторів слота 0 або 1 не проголосував за будь-який блок. Тепер припустимо, що перевага віддається лівим.

Час T_{delay} перед голосуванням чесних валідаторів у слоті 2, зловмисник використовує для голосу за *право* (блок b) від члена зловмисного комітету зі слота 1 (так зване голосування за впливом). Якщо T_{delay} добре налаштовано на поведінку розповсюдження мережі, то приблизно половина чесних членів комітету слота 2 побачать голосування, перш ніж віддати свій голос та вважатимуть *правий* лідером (через голосування за впливом) і будуть голосувати за нього. Інша ж половина побачить результат голосування лише після того, як віддасть свій голос, і, таким чином, вважає, що лівий лідирує (через результат нічиєї раніше) і голосуватимуть за нього. Тепер зловмисник спостерігає за результатом голосування, який тепер має бути розділеним до розриву $\mathcal{O}(\sqrt{W_{honest}})$. Зловмисник використовує своїх членів комітету слота 2, а також членів комітету з місцями 0 і 1, щоб балансувати голосування до

рівності. Коли нічия відновлюється, зловмисник може використовувати ту саму стратегію в наступному слоті й так далі. Варто зазначити, що зловмисник може спостерігати за результатами голосування та дізнатися, скільки чесних членів комітету вважали, що *лівi* і *правi* лідирують відповідно. І може використовувати цю інформацію, щоб покращити свою оцінку T_{delay} надалі.

Для оцінки атаки, пропонуємо розглянути подію A – мережа має затримку T_{delay}.

Подія A матиме ймовірність:

$$P(A) = P(T_{delay} = t) = \beta^t \frac{e^{-\beta}}{t!}$$

Згадаємо, що ймовірність того, що блоки у слотах k та $k+1$ створюють зловмисники:

$$P` = \beta^2$$

Тоді ймовірність того, що відбудеться атака буде:

$$P_1 = P(A) \cdot P` = \beta^{2t} \frac{e^{-\beta}}{t!}$$

Тепер розглянемо залежність атаки від часу затримки. Припустимо, що наша система цілісна і чесні валідатори становлять більшість (тобто $W_{honest} \geqslant \frac{2}{3}$). Нехай коефіцієнт зловмисників у мережі $\beta = 0.2$. Як можна побачити на рисунку 2.5 , що чим більший в нас буде час затримки, тим менша буде ймовірність. Це випливає з того, що затримка це зовнішній фактор, а отже його велике відхилення є малоймовірним. Аналізуючи дану систему, можна сказати, що час затримки $t = 2$ є найбільш успішним для нашої атаки тому, що він має достатньо велику ймовірність та гарантує, що більшість з половини користувачів, в яких мережа має затримку t, встигнуть проголосувати за цей час (припускаємо, що часу $t = 1$ буде недостатньо).

Тепер ми можемо порахувати ймовірність успіху атак за епоху

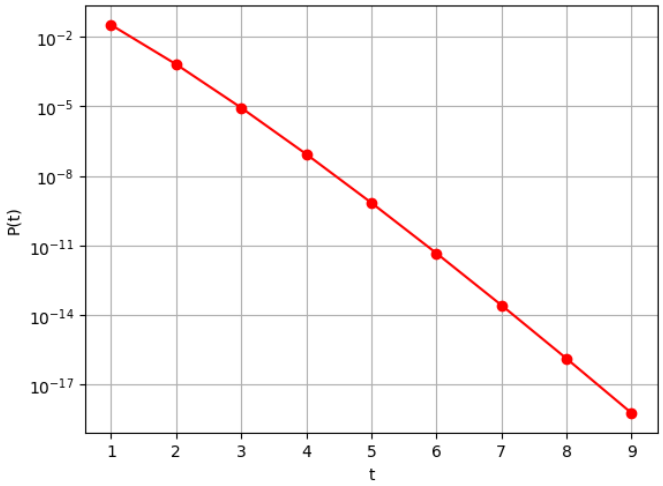

Рисунок 2.5 – Графік залежності ймовірності успіху атаки від затримки часу t. Коефіцієнт зловмисників в системі $\beta = 0.2$.

відносно кількості їх виконання. Визначимо подію B – успіх l-тої кількості атак у мережі за епоху. Тоді ймовірність такої події буде обраховуватися як l-тий добуток ймовірностей подій A:

$$P_2 = \prod_{i=1}^{l} (P_1)^i = \prod_{i=1}^{l} (\beta^{2t} \frac{e^{-\beta}}{t!})^i$$

В даній ситуації ми можемо оперувати як незалежністю цих послідовних подій, оскільки вони не впливають ніяк на валідаторів. Чесні валідатори залишаються чесними впродовж усієї епохи, як і зловмисники залишаються зловмисниками. Якщо ми підставимо значення $t = 2$, як число часу затримки яким ми можемо оперувати, то наша формула матиме вид:

$$P_2 = \prod_{i=1}^{l} (\beta^4 \frac{e^{-\beta}}{t!})^i$$

Підставляючи значення, можна переконатися, що ймовірність успіху буде експоненційно спадати, оскільки, як зазначалося вище, затримка у мережі це зовнішній вплив і тому його ймовірність з великою частотою буде близькою до 0, в чому можна переконатися розглянувши рисунок 2.6.

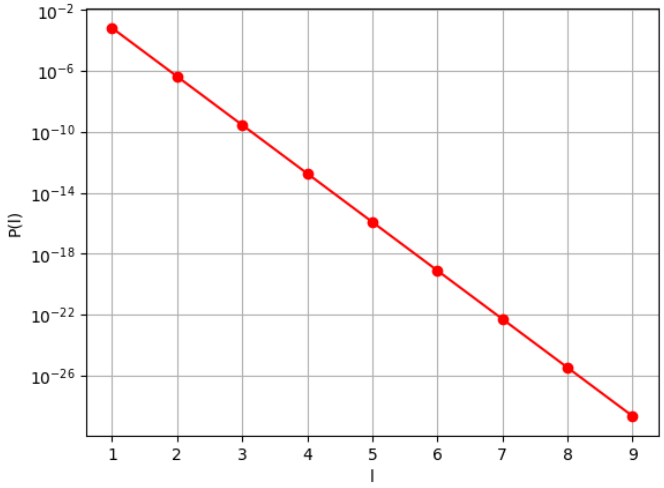

Рисунок 2.6 – Графік залежності ймовірності успіху атаки від кількості виконаних атак за епоху. Коефіцієнт зловмисників в системі $\beta = 0.2$, час затримки $t = 2$

2.3 Атака Reorg з використанням ймовірнісної мережевої затримки (Reorg attack using probabilistic network delay)

Третя атака використовує у собі комбінацію двох попередніх атак та створює умови, за яких зловмисник із невеликою часткою та без контролю затримки мережі може виконати довготривалий Reorg. Ідея полягає в тому, що зловмисник уникає конкуренції з чесними валідаторами та, використовуючи техніку удосконаленої атаки живучості, приблизно розподіляє порівну усіх чесних валідаторів, з яких одна половина бачить голосування інших до того як віддасть свій голос, а інша лише після (рисунок 2.7). Тобто можна зробити так, що чесні валідатори будуть конкурувати між собою та створювати нічию, а зловмисник може віддати голоси за ту гілку, яка йому потрібна буде. Алгоритм даної атаки виглядає наступним чином:

1) У слоті n + 1 зловмисник приватно створює блок n + 1, що буде нащадком блоку n і приватно засвідчує його, використовуючи атестацію зі

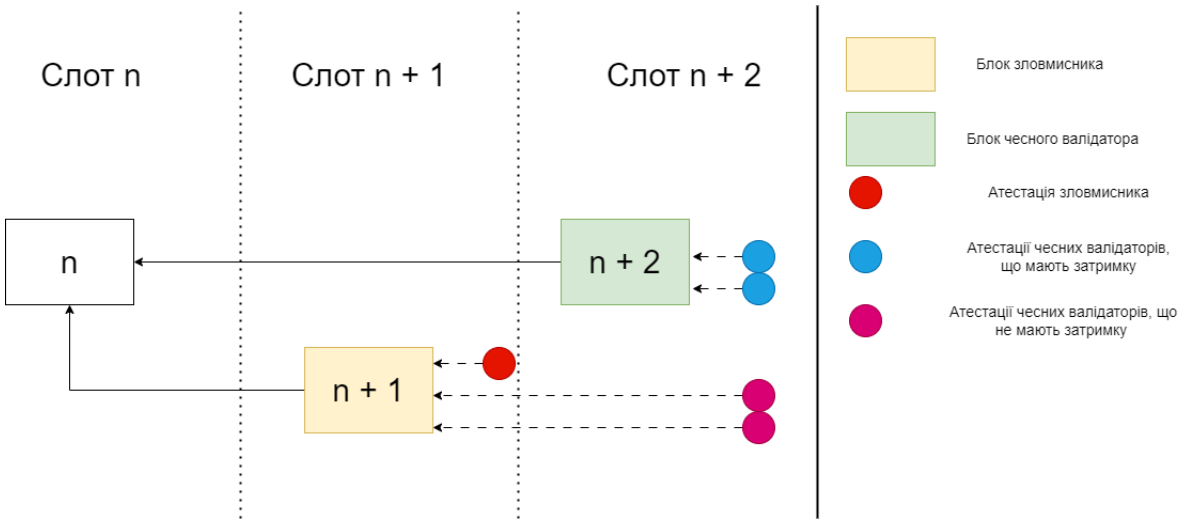

Рисунок 2.7 – Атака Reorg з використанням ймовірнісної мережевої затримки

слота n+1. Чесні валідатори не бачать блок n + 1.

2) У наступному слоті чесний валідатор створює блок n + 2, вважаючи що його предок блок n, оскільки він не бачить блок n + 1. Перед тим, як чесні валідатори будуть атестувати в слоті n + 2, зловмисник випускає блок n + 1, разом з утриманою атестацією. Таким чином, приблизно половина чесних членів комітету слота n + 2 атестує, перш ніж вони побачать вплив голосування (голосуючи за блок n + 2, як за лідера), а інша половина бачить блок n + 1 як головний через атестацію зі слота n + 1 і голосують за блок n + 1. Якщо зловмисник контролює мережеву затримку, то він може звільнити утриманий блок та проголосувати так, щоб блок n + 2 зібрав рівно на одну атестацію більше, ніж блок n + 1. Якщо мережа затримка імовірнісна, як в удосконаленій атаці живучості, тоді зловмисник повинен витратити $\mathcal{O}(\sqrt{W_{honest}})$ зловмисних голосів, щоб відновити розрив у голосах. У випадку k-reorg цей крок повторюється для перших $(k-1)$ слотів.

3) Слот n + 3 є останнім слотом атаки Reorg, а отже зловмиснику треба схилити до ланцюга зі зловмисним блоком одразу як буде створений блок чесним валідатором у даному слоті. Валідатор, що створює блок n + 3 вважає, що блок n + 2 буде контрольною точкою нашого ланцюга і тому

вважає його своїм предком. Зловмисник публікує дві останні приховані атестації так, що більшість чесних членів групи слота n + 3 переглядає їх до атестації. В результаті, більшість голосує за блок n + 1, як за останнього нащадка ланцюга.

4) У слоті n + 4, валідатор, що створює блок, розглядає блок n + 1 як кінцевий і будує блок n + 4 на блоці n + 1. Це завершує атаку 2-Reorg і знищує блоки n + 2 і n + 3.

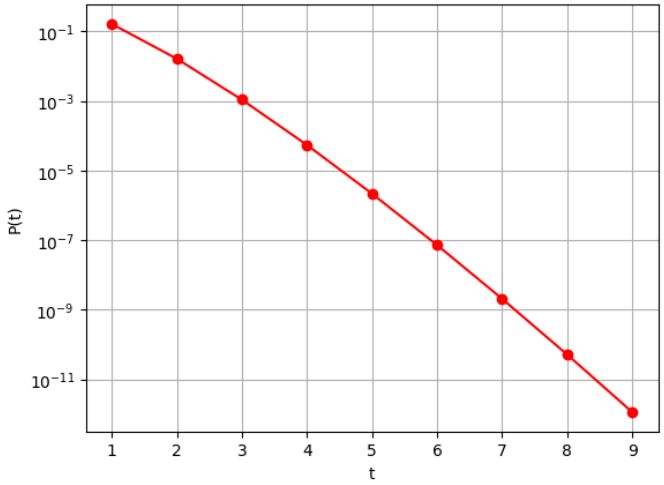

Рисунок 2.8 – Графік залежності ймовірності успіху атаки від затримки часу t. Коефіцієнт зловмисників в системі $\beta = 0.2$.

Оцінка ймовірності даної атаки є досить схожою на оцінку удосконаленої атаки живучості. Ми балансуємо між приблизною рівністю голосів за кожний блок із можливим розривом $\lfloor W_{honest} \rfloor - 1$. Тоді ймовірність події затримки часу в мережі визначається:

$$P(T_{delay} = t) = \beta^t \frac{e^{-\beta}}{t!}$$

Спираючись на результати, що зображені на рисунку 2.8, так само ймовірність буде експоненційно зменшуватися зі збільшенням часу затримки. В якості верхньої границі оберемо час затримки $t = 2$.

Згадаємо, що успіх l-тої кількості атак у мережі за епоху це добуток ймовірностей подій затримки часу в мережі, а отже загальна ймовірність

матиме вигляд:

$$P_1 = \prod_{i=1}^{l} (P')^i = \prod_{i=1}^{l} (\beta^t \frac{e^{-\beta}}{t!})^i$$

Якщо ми розглянемо графік залежності, зображений на рисунку 2.9 від l-тої кількості атак, то знову можемо побачити, що ймовірність буде експоненційно спадати з кожним кроком.

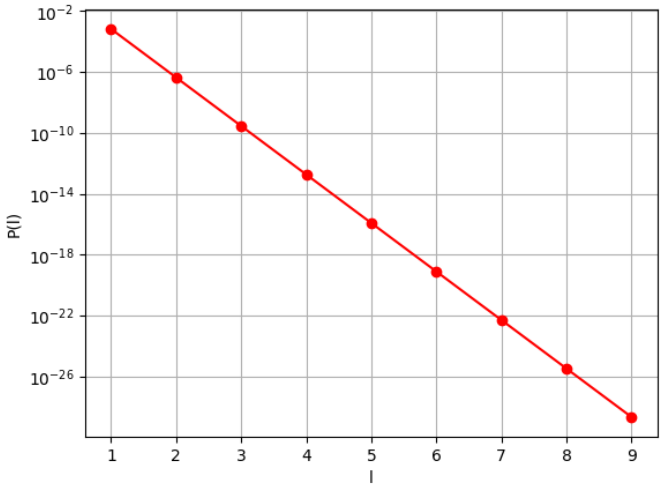

Рисунок 2.9 – Графік залежності ймовірності успіху атаки від кількості виконаних атак за епоху. Коефіцієнт зловмисників в системі $\beta = 0.2$, час затримки $t = 2$

2.4 Порівняння атак удосконаленого Reorg, живучості та Reorg з використанням ймовірнісної мережевої затримки між собою

Розглядаючи усі три вище описані атаки, можна підсумувати, що кожна з атак має як свої переваги, так і недоліки, що в результаті впливають на ймовірність успіху. Порівнюючи, наприклад між собою першу та третю атаку, що засновані на принципу Reorg, можна сказати,

що перша атака (удосконалений Reorg) має більший успіх на малих ітераціях ніж третя (Reorg з використанням ймовірнісної мережевої затримки), оскільки маніпулювання валідаторами має більшу ймовірність ніж підлаштування під час затримки у мережі. Але якщо ми говоримо

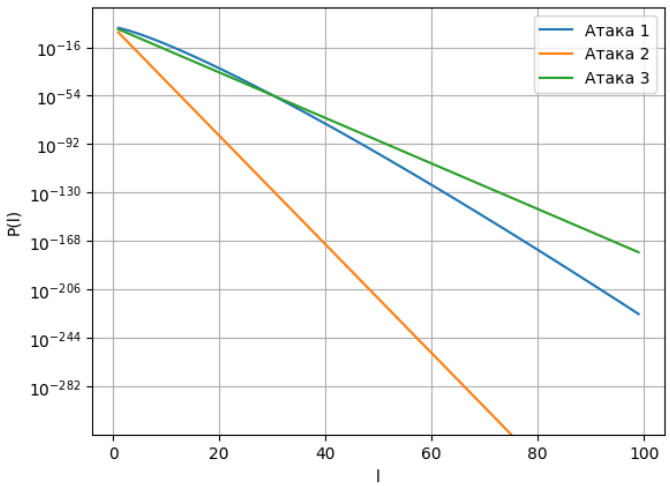

Рисунок 2.10 – Графік залежності ймовірності успіху атак від кількості виконаних атак за епоху. Коефіцієнт зловмисників в системі $\beta = 0.2$, час затримки $t = 2$

про велику кількість ітерацій, що триватиме принаймні більше ніж половина епохи, то ймовірність третьої атаки буде більшою за першу, оскільки чесний валідатор не може голосувати за нечесний блок, бо це буде порушення умов, що ми накладаємо. Відповідно, з кожною ітерацією більше валідаторів розумітиме, що блок, який притримав зловмисник є нечесним, а отже не будуть віддавати за нього свої атестації.

З приводу другої атаки (атака живучості) варто розуміти, що вона програє іншим атакам через малу ймовірність очікування сприятливих умов, а саме те, що два зловмисники будуть підряд створювати блоки. Інакше б вона мала схожу ймовірність до атаки Reorg з використанням ймовірнісної мережевої затримки. Порівняння ймовірності успіху усіх трьох атак можна побачити на рисунку 2.10.

Варто нагадати, що було розглянуто оцінки для цілісної системи та

для невеликого коефіцієнта зловмисників в мережі. Це було необхідно, щоб показати, як працюватимуть ці атаки у нормальній системі, яка прописана у самому протоколі консенсусу. Звичайно бувають різні ситуації, але це вже окремі випадки, які матимуть або ймовірність $P = 1$, якщо зловмисники будуть становити більшість, або припустити, що часу затримки $t = 1$ буде достатньо для атестації блоку валідаторами (рисунок 2.11).

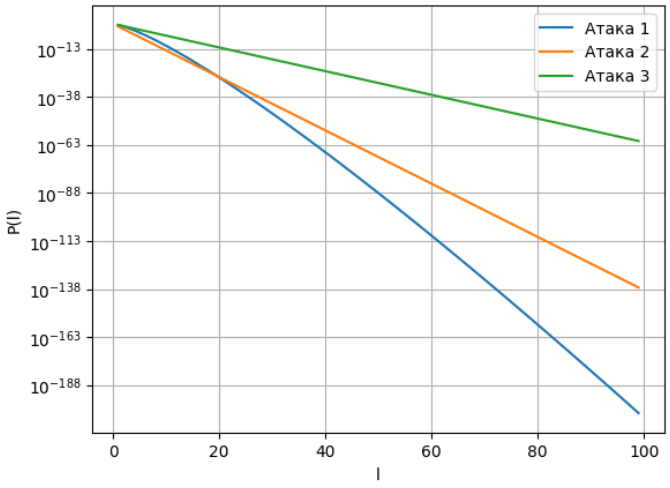

Рисунок 2.11 – Графік залежності ймовірності успіху атак від кількості виконаних атак за епоху. Коефіцієнт зловмисників в системі $\beta = 0.34$, час затримки $t = 1$

Висновки до розділу 2

Підсумовуючи вище отримані оцінки можемо зробити висновок, що протокол Casper FFG є доволі стійким до цих атак централізації та саботажу. Хоча варто додати, що ймовірність однієї атаки за епоху є недостатньо малою, щоб вважати, що атаки не матимуть потенційного успіху. Найбільш загрозливою атакою, спираючись на розрахунки, є Reorg з використанням ймовірнісної мережевої затримки, оскільки він комбінує у собі простоту виконання (потрібен лише один зловмисник, що створюватиме блок) та розподіл голосів за мережевою затримкою, що є

більш успішним ніж маніпуляція чесними валідаторами в удосконаленій атаці Reorg.

ВИСНОВКИ

В результаті проведеного дослідження було детально розглянуто принципи роботи протоколу консенсусу Casper FFG. Його було обрано в якості об'єкта дослідження, оскільки він використовується однією з найвідомішою платформою блокчейн Etherium.

Даний протокол працює за алгоритмом DPoS, що є удосконаленою версією алгоритму PoS. Завдяки цьому, ми можемо гарантувати, що кожної епохи в нас будуть відбуватися чесний відбір валідаторів, що створюють блоки й таким чином запобігати можливим умовам, коли успішність атак на протокол буде тривати більше однієї епохи підряд.

Також було розглянуто дві атаки, до яких даний протокол є вразливий: emphдовгострокові перегляди (long range revisions) та *катастрофічні збої (catastrophic crashes)*. Атака довгострокових переглядів виконується під час зміни валідаторів у системі. Оскільки в цей момент їх депозити вже були виведені з цієї системи, але самі валідатори ще можуть оперувати із блоками, вони можуть підтвердити другий блок, залишаючи систему з двома конфліктними точками на які розбився ланцюг. При цьому варто зазначити, що валідатори є порушниками, але вони не будуть покарані, оскільки їх рахунок депозитів є нульовим. Атака катастрофічних збоїв показує нам, що буде, якщо не працює алгоритм BFT. Якщо більшість валідаторів вийде із ладу, то відповідно система не буде працювати. Цим атакам можна запобігти, відповідно налаштувавши правила консенсусу, не сильно видозмінюючи сам протокол.

На відміну від попередніх атак, нові атаки, що були розглянуті у другому розділі не мають комплексного рішення для запобігання. В даній роботі було досліджено оцінки успіху цих атак та, як результат, стійкість протоколу до них. В якості обрахунку ймовірностей було обрано

пуассонівський розподіл, оскільки він добре описує велике число подій, кожна з яких рідко зустрічається. Це відповідає нашій системі, оскільки в нас створюється велике число блоків і, відповідно, велика кількість подій, яка може з ними відбутися. Оскільки валідатори в системі змінюються кожної епохи, було розглянуто ймовірності успіху атак впродовж цієї одиниці, при цьому розглядаючи можливість виконання атаки аж до 100 разів.

Для кожної атаки (удосконалений Reorg, удосконалена живучість та Reorg з використанням ймовірнісної мережевої затримки) було виведено оцінки ймовірності успіху для l-тої кількості виконаних атак за епоху. Окрім цього, для двох останніх було також побудовано оцінки ймовірності часу затримки та проаналізовано для якого часового проміжку найкраще буде обраховувати ймовірність довготривалих атак. Отримані оцінки були порівняні між собою і визначено, що найбільшу ймовірність для довготривалих атак має атака Reorg з використанням ймовірнісної мережевої затримки, на малих ітераціях – удосконалений Reorg.

Ґрунтуючись на отриманих оцінках, можна підсумувати, що протокол Casper FFG є доволі стійким до нових атак централізації та саботажу удосконаленого Reorg, живучості та Reorg з використанням ймовірнісної мережевої затримки між собою. Оскільки з кожним наступним кроком виконання атак впродовж епохи, їх ймовірність успіху експоненційно спадає, можна зробити висновок, що принаймні 90% ланцюга не буде саботовано та система буде справно працювати, не порушуючи принцип децентралізації.

В якості наступних кроків необхідно більш детально дослідити проблему виникнення цих атак, для уникнення їх надалі. Хоч і було стверджено, що алгоритм стійкий до тривалих атак, але на жаль, ймовірність хоча б однієї атаки за епоху є досить значною, щоб вважати її виконання можливим. Також важливо досліджувати інші атаки, що з'являються досить швидко останнім часом, оскільки проблема централізації та саботажу мають досить великі наслідки для мережі. І

якщо протокол буде вразливий до них, зокрема до тривалого виконання атаки, то це стане сигналом, що даний протокол не буде надійним для підтримки консенсусу в системі.

ПЕРЕЛІК ПОСИЛАНЬ

[1] “Byzantine Fault Tolerance Explained”. В: *University journal crypto.com* (6 груд. 2018). URL: https://academy.binance.com/en/articles/byzantine-fault-tolerance-explained.

[2] Akshita Jain та ін. “Proof of stake with Casper the friendly finality gadget protocol for fair validation consensus in Ethereum”. В: *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 3 (3 2018).

[3] “PROOF-OF-STAKE (POS)”. В: *Ethereum Foundational topics* (2023). URL: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/.

[4] Ashish Rajendra Sai та ін. “Taxonomy of centralization in public blockchain systems: A systematic literature review”. В: *Information Processing and Management* 58 (4 2021). ISSN: 03064573. DOI: 10.1016/j.ipm.2021.102584.

[5] Caspar Schwarz-Schilling та ін. “Three Attacks on Proof-of-Stake Ethereum”. В: т. 13411 LNCS. 2022. DOI: 10.1007/978-3-031-18283-9_28.

[6] “What Is Delegated Proof of Stake?” В: *University journal crypto.com* (2022). URL: https://crypto.com/university/what-is-dpos-delegated-proof-of-stake.

[7] V Zamfir та ін. “Introducing the minimal CBC Casper family of consensus protocols”. В: *DRAFT v1. 0* 5 (2018).

[8] Leslie Lamport та ін. “The Byzantine Generals Problem”. В: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4 (3 1982). ISSN: 15584593. DOI: 10.1145/357172.357176.

[9] Michael Neuder та ін. “Low-cost attacks on Ethereum 2.0 by sub-1/3 stakeholders”. Англ. В: 3 лют. 2021. arXiv: 2102.02247v1 [math.LO].

[10] Vitalik Buterin та ін. *Combining GHOST and Casper*. Англ. 11 трав. 2020. arXiv: 2003.03052v3 [math.LO].