ВСТУП

Основними тенденціями розвитку сучасних інформаційних технологій та їх систем кібербезпеки є: подальший розвиток розподіленого оброблення інформації, реалізованого в сучасних грід- та хмарних інформаційно-телекомунікаційних системах; поява нових моделей обчислень та їх практична реалізація (квантові обчислення і квантова криптографія); міграція принципів побудови інформаційно-комунікаційних систем загального призначення у галузь автоматизованих систем управління критичною інфраструктурою; перехід від інформаційно-комунікаційних систем до кіберпростору, тобто від безпосередньо керованого інформаційно-комунікаційного середовища до децентралізованих систем управління, спроможних надати поведінці деякої області кіберпростору «інтелектуальний» характер.

Кіберпростір (приставка «кібер» означає «управління») – це система, в якій самостійно виникають сигнали, що забезпечують керування процесами збереження певного стану системи. Тоді інформаційна система повинна мати здатність до самоорганізації та децентралізації та бути розподіленою за функціями та ресурсами. Структури даних і процеси, які використовуються в системі, мають відповідати принципам функціонування системи, саме тому тенденція децентралізації управління інформаційно-комунікаційними сервісами зумовила виникнення технології блокчейн та її «похідних» — криптовалют, інтелектуальних контрактів, тощо. Технологія блокчейн є основою розподіленої децентралізованої захищеної технології оброблення інформації, призначеної для розв’язання широкого кола прикладних задач — від децентралізованого випуску та обігу електронної готівки (криптовалюти), аутентифікації та електронного нотаріату до розподіленого підписання контрактів і проведення електронних виборів. З іншого боку, для ефективного використання технології блокчейн потрібно розв’язати низку теоретичних та практичних задач, зокрема:

1. Розроблення та вдосконалення протоколів узгодження в

розподілених ненадійних системах.

2. Дослідження моделей даних у блокчейн-системах. Традиційною для технології блокчейн є модель збереження даних із максимальною надлишковістю, при якій на кожному вузлі мережі зберігається повна копія бази даних. Альтернативною моделлю збереження даних є модель мінімальної надлишковості, при якій частки бази даних розподіляються між всіма вузлами блокчейну і для відновлення повної бази даних необхідна участь всіх вузлів. Між цими двома граничними випадками існують інші моделі збереження даних із різними ступенями надлишковості. Актуальною є задача побудови механізму реплікації таких розподілених баз даних при різних ступенях надлишковості збереження даних.

3. Модифікація наявних криптографічних механізмів блокчейну, наприклад, заміна дерев Меркла, використання групових цифрових підписів, заміна задачі пошуку прообразу геш-функцій на інші складні для обчислення задачі.

4. Аналіз стійкості геш-функцій до обернення з врахуванням особливості їх використання в блокчейні, який відрізняється від традиційного аналізу колізій стійкості та обернення (пошуку прообразу) геш-функцій.

5. Аналіз та побудова клептографічних лазівок блокчейну.

6. Побудова криптографічних протоколів на основі блокчейнів, наприклад, протоколів доказу з нульовими знаннями, анонімізації, доказу інтелектуальної власності тощо.

7. Розрахунки параметрів практичних блокчейн-систем (оцінювання надійності реальних блокчейнів, довжини буферів транзакцій, що очікують включення в блокчейн та ін.).

В навчальному посібнику надані теоретичні основи функціонування технології блокчейн та криптовалют, аспекти їх практичного використання, переваги та недоліки зазначених технологій. Частина 1 присвячена теоретичним засадам функціонування блокчейн-технологій, аналізу атак на

блокчейни, строгому математичному обгрунтуванню вибору параметрів блокчейну. В першу частину не війшли всі аспекти практичної реалізації блокчейн-технологій (аспекти створення програмних додатків на смарт-контрактах, опис блокчейн-платформ та ін.), цьому присвячена 2 частина посібника.

1. ПРОТОКОЛИ КОНСЕНСУСУ У ТЕХНОЛОГІЇ БЛОКЧЕЙН. ПОРІВНЯЛЬНИЙ АНАЛІЗ ТА РІЗНІ ТИПИ АТАК

За декілька останніх років кількість людей, що в своїй діяльності зустічає поняття блокчейну та криптовалюти, швидко зростає. Криптовалютна лихоманка охопила певний сегмент ринку, вона зростає та спадає хвилями. Сьогодні криптовалюти стали глобальним явищем, до якого стають причетними все більше людей та організацій.

Слід зазначити, що більшість людей помилково ототожнюють поняття блокчейну та криптовалюти. Насправді ж потенційна сфера використання блокчйну є надзвичайно широкою, а використання блокчейн-технології для побудови криптовалюти – лише один з її багатьох можливих застосувань.

Як видно з назви, при побудові криптовалюти використовуються різні криптографічні механізми – цифрові підписи, геш-функції, навіть протоколи доведення без розголошення. Стійкість цих механізмів уже багато років аналізувалась і доводилась, тому атаки на блокчейн, як правило, не пов’язані з атаками на них. Найбільш поширеними атаками є атаки на сам блокчейн – в першу чергу, на протокол консенсусу, який лежить в його основі.

У цьому розділі розглянемо основні поняття, пов’язані з блокчейном та криптовалютами. Також проаналізуємо алгоритми консенсусу, що використовуються у сучасних блокчейнах, і приділимо особливу увагу найбільш поширеним з них – Proof-of-Work (PoW) та Proof-of-Stake (PoS). Після аналізу алгоритмів консенсусу будуть наведені різні типи атак на блокчейн, серед яких найбільш небезпечними є атака підміни блока (АПБ) (у випадку криптовалютного блокчейну така атака називається атакою подвійної витрати, АПВ) та атака розгалуження (АР).

1.1. Історія появи криптовалюти та блокчейну

Мало хто знає, що криптокультура виникла як побічний продукт іншого винаходу. Сатоши Накамото, невідомий винахідник Біткойну ніколи

не мав наміру вигадувати цифрову валюту. У своїй першій і єдиній роботі, яка з’явилась наприкінці 2008 року, Сатоши написав, що він розробив "електронну однорангову систему грошової одиниці". Зокрема, він зазначив, що завдяки цій системі можливо захиститись від атаки подвійної трати коштів, яка поширена в інших грошових онлайн системах. Він був першим, кому вдалось побудувати повністю децентралізовану мережу без сервера та централізованого керування. Але історія також пам’ятає інші, менш вдалі, спроби інших винахідників.

Паливні картки

Одна з перших спроб створити цифрову валюту передувала появі Біткойн приблизно на 20 років. Автозаправочні станції в Нідерландах постійно страждали від нічних крадіжок. Замість того, щоб посилювати охорону і ризикувати життями людей, команда розробників вирішила відмовитися від грошей і замінити їх спеціальними пластиковими картами. Водії платили цими картами замість готівки, і необхідність в зберіганні великих сум відпала. Це один з перших прикладів створення електронних грошей, і він має пряме відношення до сучасних цифрових валют.

DigiCash

Приблизно в той же час американський криптограф Девід Чаум експериментував з іншою формою електронних грошей – токенами. З їх допомогою можна було проводити безпечні і конфіденційні платежі. Чаум розробив так званий сліпий цифровий підпис для забезпечення цілісності та конфіденційності інформації, якою обмінюються сторони. Він забезпечував перевірку справжності і можливість модифікації без відстеження. Через кілька років Чаум заснував DigiCash. Компанія збанкрутувала в 1998 році, проте концепція, формули і криптографічні протоколи зіграли важливу роль у розвитку цифрових валют.

PayPal

У 1990-х різні стартапи намагалися розвинути і вдосконалити концепцію DigiCash. З них, мабуть, найвідоміший – PayPal. Свого часу

компанія зробила революцію – онлайн платежі. Вона дозволила всім охочим швидко і безпечно переказувати гроші через Інтернет. Уклавши угоду з eBay, PayPal отримала доступ до величезної бази користувачів, що дозволило їй рости швидкими темпами. До цього дня PayPal залишається одним з найбільших платіжних сервісів і є прикладом для нащадків (в тому числі для компаній, які намагалися налагодити торгівлю золотом через інтернет). Одним з найвідоміших послідовників PayPal став сервіс e-gold, який пропонував приватним особам кредитування під заставу золота та інших дорогоцінних металів. Пізніше він був закритий федеральною владою США в 2005 році.

B-Money

У 1998 році Вей Дай розробив «анонімну, розподілену систему електронних грошей» під назвою B-Money. Дай запропонував два різних протоколи. Один з них вимагав синхронного і завадостійкого каналу зв'язку. Концепція B-Money досить сильно відрізнялась від Bitcoin і розвитку не отримала. Однак це була важлива спроба створити анонімну, приватну і безпечну електронну систему. В системі B-Money для переказу валюти за допомогою децентралізованої мережі використовувалися цифрові псевдоніми. Система передбачала автономне виконання контрактів без залучення третьої сторони. Вей Дай опублікував повноцінне дослідження про свою валюту, проте вона не змогла залучити достатню увагу для успішного запуску. У своїй роботі Накамото посилається на деякі елементи B-Money, тому вплив ідей Вей Дая на нинішні криптовалюти незаперечний.

Bit Gold

Система електронних грошей Bit Gold (не слід плутати із сучасною біржою з такою ж назвою) виникла приблизно в той же час, що і B-Money. Її творцем став Нік Сабо; робота системи була заснована на власному алгоритмі доведення виконання роботи (Proof-of-Work), який, деякою мірою, повторює процес майнінгу біткойнів. Процес багато в чому нагадував роботу сучасного блокчейну. Мабуть, найбільш революційним аспектом Bit Gold

стала відмова від централізованого статусу. Bit Gold прагнув уникнути залежності від єдиних засобів розподілу і управління. Сабо мав намір створити цифровий аналог золота і повністю виключити посередників. На жаль, як і B-Money, Bit Gold зазнав невдачі. Однак проект став джерелом натхнення та ідей для багатьох цифрових валют, що з'явилися на ринку через десятиліття.

Hashcash

Розроблена в середині 1990-х, Hashcash була однією з найуспішніших цифрових валют до появи Bitcoin. Вона була створена для різних цілей, в тому числі мінімізації поштового спаму і запобігання DDoS-атак. Hashcash відкрила широкий спектр можливостей, більшість з яких були реалізовані тільки два десятиліття по тому. Як і сучасні криптовалюти, Hashcash використовувала алгоритм доведення роботи для генерації і розподілу нових монет. У 1997 році, зіткнувшись з недоліком обчислювальної потужності, вона почала втрачати ефективність і в кінцевому підсумку припинила існування. У дні свого розквіту Hashcash користувалася великим попитом і інтересом. Багато з її принципів були успадковані у системі Bitcoin.

Поява Bitcoin

Поява Bitcoin в 2009 році призвела до виникнення нового покоління цифрових валют. Основними відмінностями Bitcoin стали його децентралізація і використання блокчейну. Накамото намагався побудувати цифрову грошову систему без центрального органу як однорангову мережу для обміну файлами. Це рішення стало народженням криптовалюти.

Для створення цифрової валюти потрібна мережа платежів з рахунками, балансами та транзакціями. При цьому однією з основних проблем, яку має вирішити кожна платіжна онлайн-система, є запобігання так званим подвійним витратам, коли один суб'єкт системи витрачає одну й ту ж суму двічі. Зазвичай це робиться центральним сервером, який реєструє баланс, але в децентралізованій мережі такого серверу немає. Отже, для виконання цієї роботи потрібно, щоб кожен учасник мережі мав список всіх

транзакцій та можливість перевірити кожну з них, для протидії різним атакам, у тому числі АПВ. Якщо вузли мережі не згодні хоча б з одним балансом, все зламається – для коректного функціонування потрібен абсолютний консенсус. У централізованих системах при різних конфліктних або суперечливих ситуаціях рішення приймає уповноважений центральний орган. Але як можливо досягти консенсусу без центрального органу? До створення Bitcoin ніхто не вірив, що це навіть можливо.

1.2. Визначення блокчейну як децентралізованої системи обробки інформації

Для розгляду поняття протоколів консенсусу або узгодження потрібно зазначити, що блокчейн є системою децентралізованої обробки (збереження, додавання, модифікації) інформації. При цьому потрібно наголосити, що рішення про обробку інфомації приймається більшістю учасників системи, сформованою за деяким правилом.

Визначення. Блокчейн – система децентралізованої оброблення інформації, в якій рішення про обробку приймається за результатами голосування більшості учасників системи в ході виконання протоколу консенсусу (протоколу узгодження),

Чому тоді таку систему називають «блокчейном»? Вся справа в тому, що всі зміни в системі відбуваються в дискретні моменти часу – з точністю до «транзакції».Транзакція є найменшою можливою зміною (атомарною зміною) стану системи. Транзакції групуються в блоки; блоки утворюють впорядкований набір або ланцюг блоків, який і називають блокчейном (від англ. blockchain). Для організації та впорядкування блокчейну, кожен блок містить посилання на попередній блок. Найперший блок в ланцюзі називається початковим блоком або блоком генезису (від англ. genesis block). Він не має посилання на попередній блок і, як правило, є закодованим безпосередньо в протокол. Нові блоки додаються до системи відповідно

визначеному набору правил, встановлених протоколом, який і називається протоколом консенсусу.

Важливою функцією цих правил є захист від атак зловмисників на блокчейн і досягнення консенсусу у випадку появи декількох варіантів ланцюга блоків.

1.3. Протоколи консенсусу та їх характеристики

По відношенню до блокчейну, консенсусом називають певну процедуру, яка має риси процедур голосування та узгодження. Мета цієї процедури – щоб усі учасники блокчейну (принаймні чесні учасники) прийшли до певного однакового бачення інформації, що зберігається в блокчейні.

Отож, протоколи консенсусу – це сукупність правил, які допомагають:

- забезпечити виконання процесу обробки інформації в блокчейні;
- усунути неоднозначність продовження ланцюга блоків блокчейну (генерацію розгалуження ланцюга блоків);;
- перевіряти чесність учасників.

Консенсусний протокол має наступні складові:

- правила створення наступного блоку;
- правила вирішення конфліктних ситуацій, які можуть виникати в мережі як ненавмисне, так і внаслідок порушення правил нечесними учасниками мережі;
- правила заохочення учасників мережі для підтримки функціонування блокчейну та виконання даного протоколу.

Отже в подальшому протоколом узгодження ми будемо називати інтерактивний (тобто такий, в якому задіяний більше ніж один учасник) алгоритм, який завершується обчисленням спільного для всіх учасників значення.

Проблема побудови протоколів узгодження у розподілених мережах

була відома задовго до появи поняття «блокчейн» як проблема «візантійських генералів» (або «візантійських угод») та формулювалась як необхидність побудови наступного інтерактивного алгорітму.

Нехай є n учасників алгоритму. Тоді алгоритм завершується за кінцеву кількість кроків, якщо для будь-якого початкового входу x_i, $i = 1, \ldots, n$ і-того учасника та деякого параметра d мають виконуватися такі умови:

Умова завершення. Всі «чесні» учасники обчислень в кінці протоколу набувають значення d.

Умова коректності. Якщо існує значення x, таке, що для усіх «чесних» учасників $x = x_i$, тоді $d = x$.

Відомо, що такий алгоритм буде завершуватись за кінцеву кількіть кроків якщо загальна кількість учасників повинна перевищувати кількість «нечесних» учасників більше ніж у три рази,$n > 3 \cdot f$, де f — кількість «нечесних» учасників. Зрозуміло, що такі умови практично унеможливлюють пряме використання зазначених протоколів. Візьмемо до уваги, що більшість таких протоколів створювалась для розподілених баз даних в мережах із відомою архітектурою, без врахування асинхронності та невизначеності архітектури сучасних глобальних інформаційно-телекомунікаційних систем. Тому при побудові нових асинхронних протоколів узгодження повинно враховувати такий фактор як неможливість передбачити кількість активних в даний час вузлів мережі. Це визначає необхідність здійснення випадкового вибору учасника протоколу та створення механізму забезпечення цілісності всіх попередніх транзакцій (далі – «механізму цифрового пломбування»). Додатковою вимогою є створення механізму «заохочення» участі у протоколі для гарантування його працездатності.

Зважаючи на це, основними ідеями створення протоколу узгодження в блокчейнах (на прикладі блокчейна криптовалюти Bitcon) стали:

забезпечення синхронізації за допомогою ланцюга блоків транзакцій;

реалізація механізму цифрового пломбування за рахунок цифрового підпису, заснованого на ідентифікаторах;

реалізації загальноприйнятих для мережі правил генерації наступного блоку транзакцій в ланцюзі блоків;

забезпечення сталої значної обчислювальної складності вирішення задачі генерації наступного блоку транзакцій в ланцюзі блоків;

забезпечення заохочення участі учасників протоколу у вирішенні задачі генерації наступного блоку транзакцій в ланцюзі блоків;

можливості перевірки правильності генерації наступного блоку транзакцій в ланцюзі блоків будь-яким учасником протоколу.

Розглянемо протокол узгодження блокчейну криптовалюти Bitcoin більш детально тому що:

це перший з відомих «справжньо» децентралізованих протоколів узгодження,

цей протокол насьогодні використовується в інших протоколах узгодження на етапі ініціалізації.

1.3.1 Сутність протоколу консенсусу Proof-of-Work

Основною ідеєю цього класу протоколів є використання «складної» обчислювальної задачі як інструменту організації змагання за право згенерувати наступний блок. Отже основним цинним ресурсом, виміром, за яким визначається переможець – швидкість ришення цієї складної задачі. Для вирівнювання шансів учасників протоколу та надійності функціонування системи для рішення цієї задачі не повинно існувати швидкого алгоритму її виришення (алгоритму поліноміальної обчислювальної складності). Тому як складну задачу обирають зазвичай таку, для рішення котрої відомий тільки алгоритм прямого перебору.

Мабуть, найвідомішим таким протоколом можна вважати повільний, але надійний протокол узгодження Proof-of-Work криптовалюти Bitcoin: кожен вузол мережі, сформувавши новий блок, підбирає деяке число nonce, що є одним з параметрів блоку. Блок вважається дійсним, якщо виконується

валідність всіх транзакцій, які він підтверджує, та геш-функція від цього блоку має таке значення, що відповідне йому десяткове число є меншим за деяке наперед задане цільове (target) значення T: $H\left(h_{i-1} \| tr \| nonce\right) \le T$. Наприклад, значення геш-функції повинно починатись з деякої кількості нулів (на 2019 рік - це 62-64 нулі). Отже як складна задача тут використовується задача знаходження прообразів геш-функції певної довжини, для вирішення якої насьогодні відлмий тількі алгоритм прямого перебору по всім значенням $nonce$.

Proof-of-Work (PoW – дослівно: доведення виконання роботи) – алгоритм захисту розподілених систем від зловживань (DDoS-атак, спам-розсилок тощо), сутність якого зводиться до трьох основних пунктів:

- необхідності повільного виконання певного, досить складного, завдання;
- можливості швидко і легко перевірити результат;
- неможливості наперед спрогнозувати, хто саме створить наступний блок, але імовірність створення наступного блоку певним майнером пропорційна його обчислювальним потужностям (точніше – дорівнює частці його обчислювальних потужностей, або його *гешрейту*).

Вперше концепція Proof-of-Work була описана в 1993 році в роботі «Pricing via processing or combatting junk mail, advances in cryptology». І хоча сам термін в статті ще не використовувався, автори запропонували наступну ідею: «Щоб отримати доступ до загального ресурсу, користувач повинен обчислити деяку функцію: досить складну, але посильну; так можна захистити ресурс від зловживання».

У 1997 році Адам Бек запустив проект Hashcash, присвячений захисту від спаму. Завдання формулювалося так: «Знайти таке значення x, що значення геш-функції $\mathrm{SHA}(x)$ містить N старших нульових біт». Для досягнення мети потрібно було 252 геш-обчислень. І якщо для відправки кількох звичайних листів додаткові розрахунки перешкод не створюють, то необхідність постійного перерахунку робить відправку спаму дуже

ресурсномісткою. При цьому перевірка коректності обчисленого здійснюється дуже швидко: використовується одне обчислення SHA із заданим аргументом.

У 1999 році з'являється і сам термін Proof-of-Work – використаний він був у статті «Proofs of Work and bread pudding protocols» в журналі «Communications and multimedia security».

Роль алгоритму Proof-of-Work у блокчейні системи Bitcoin

У мережі Bitcoin механізм PoW був використаний як засіб досягнення консенсусу. При цьому за основу Сатоши Накамото взяв ідею згаданого вище проекту Hashcash, додавши до нього механізм змінення складності – зменшення або збільшення цільового значення T або кількості N нулів у геш-значення. Обчислювальною функцією стала SHA-256.

Якщо говорити простими словами, механізм PoW забезпечує здатність вузла мережі (ноди) перевірити, що майнер (добувач криптовалюти), в ролі якого виступає вузол, який додає новий блок в блокчейн, фактично виконав розрахунки. Даний процес включає в себе спробу знайти хеш заголовка блоку, який буде за своїм значенням відповідати поточному рівню складності.

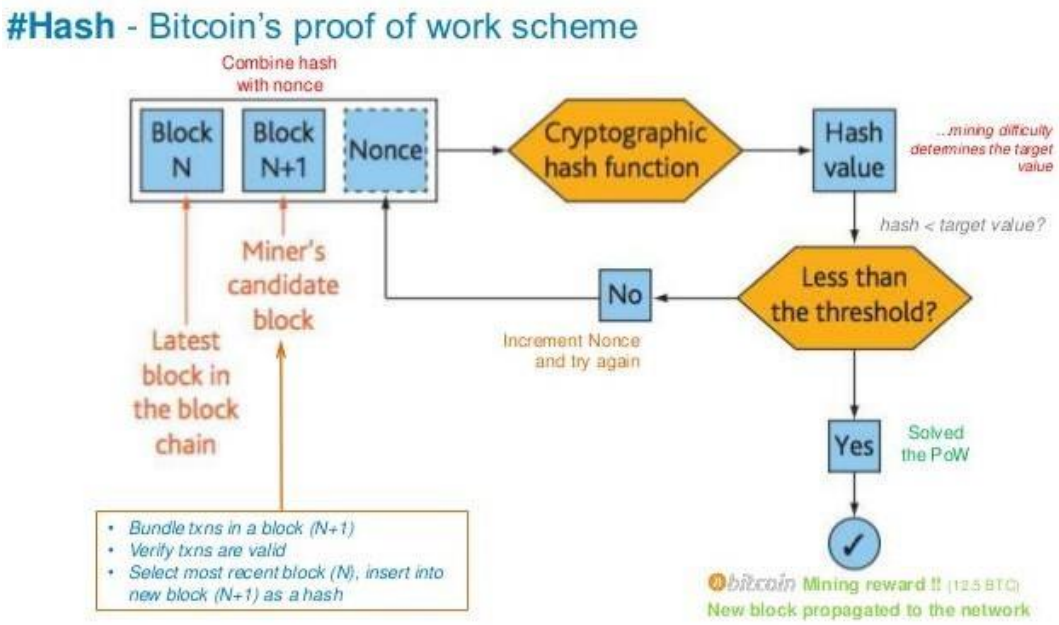
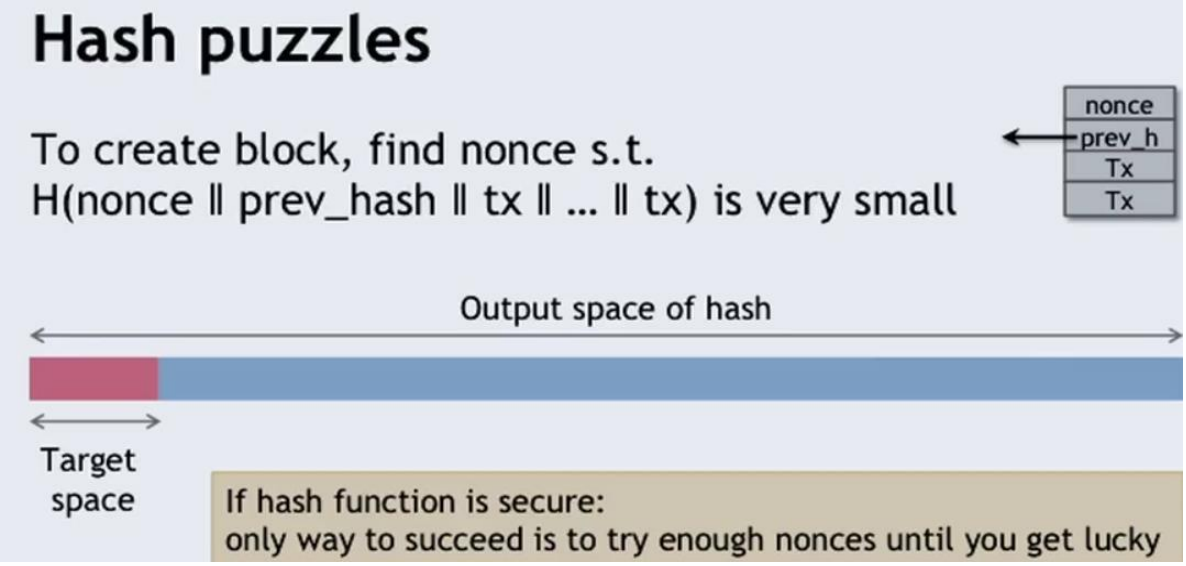


Схема роботи Proof-of-Work

Розглянемо, як саме працює Proof-of-Work. Задача кожного вузла (претендента на формування блоку) вирішити хеш-загадку. Йому потрібно підібрати значення nonce (один з атрибутів блоку) так, щоб геш усього блоку (а це й список транзакцій, й хеш попереднього блоку, й саме число nonce) отримало потрібне значення (визначене кількістю нулів на початку; тобто результат повинен потрапити в малу, порівняно з усією вихідною областю, зону). І тому вважаємо, що якщо геш-функція є криптографічною, то єдиний спосіб досягти успіху у вирішенні цієї геш-головоломки – це просто перебирати значення nonce до тих пір, поки результат не потрапить у потрібну область. Тож конкретно, якщо множина можливих значень буде лише 1% загального простору, доведеться спробувати близько 100 значень, перш ніж пощастить. Тому кожну мить вузли в системі, конкуруючи, обраховують значення nonce у сформованих блоках для того, щоб отримати винагороду. Ось чому система криптовалюти децентралізована – ніхто не обирає формувача блоку.



Основними властивостями механізму Proof-of-Work є великі

обчислення, їх швидка перевірка та параметризована вартість.

Великі обчислення

З часом складність розрахунків змінюється таким чином, щоб у середньому 1 блок формувався 10 хвилин. Нагорода майнеру за обчислений блок також змінюється (наприклад, на 2018 рік це 12,5 BTC за сформований блок у мережі Біткойн). Станом на серпень 2014 року для генерації блоку потрібно було в середньому обрахувати 2000 пента-хешів, а у червні 2018 року потужність мережі Біткойн складала 40 екзо-хешів. Звичайний ноутбук або навіть одиничний суперкомп'ютер ніколи не зможуть конкурувати з фермами та цілими майнінг-пулами задля успіхів у системі.

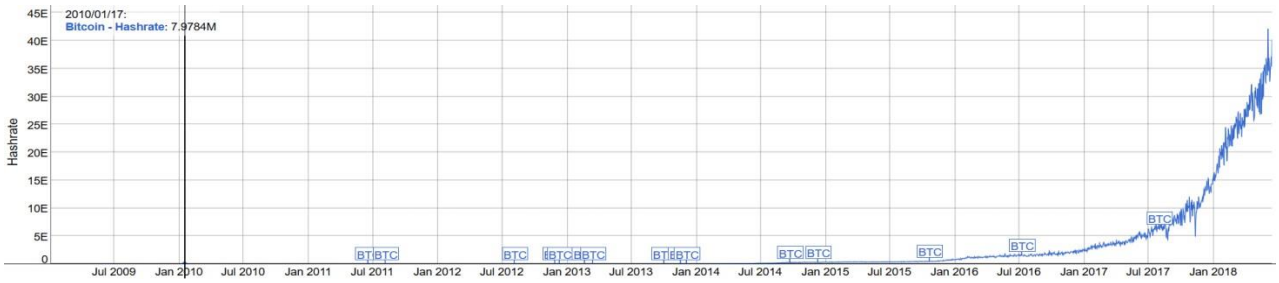
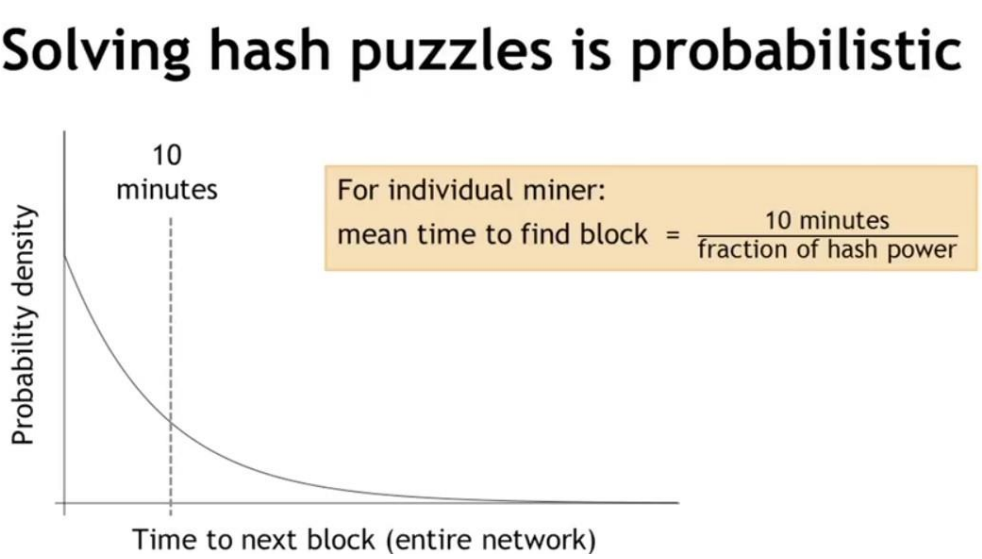


Рис.- 1.3. Зростання потужності майнерів.

Вирішення хеш-загадки є ймовірнісним – ніхто не може передбачити, який nonce призведе до відповіді. Тому є невелика ймовірність того, що наступний блок буде знайдено навіть через кілька секунд. І є також невелика вірогідність того, що це займе багато часу, може, годину.

Для конкретного майнера середній час знаходження блоку, 10 хвилин, які буде розділено на частку потужності гешу, яку він контролює. Отже, якщо майнер має 0,1% загальної хеш-потужності мережі, він знайде блок приблизно раз за 10 000 хвилин.



Швидка легка перевірка

Навіть якщо вузол обрахує мільярд хешів, щоб сформувати блок, правильне значення nonce, котре він знайшов методом перебору і підстановки до функції хешування, буде записано у новий блок. Тому для будь-якого іншого вузла тривіальним є перегляд вмісту блока, хешування всіх їх разом і перевірка того, що вихідний текст є меншим за ціль.

Це важлива властивість, яка також дозволяє позбутися централізації. Системі не потрібні централізовані повноваження, які підтверджують, що майнери виконують свою роботу правильно, бо будь-який вузол самостійно може негайно перевірити, що блок, знайдений іншим майнером, задовольняє цю властивість Proof-of-Work; інші ж учасники мережі можуть бути впевнені, що конкретний майнер дійсно задіяв великі обчислювальні потужності для знаходження конкретного блоку.

До основних недоліків протоколу PoW можна віднести

по-перше, високі витрати.

- Для майнінгу потрібне спеціальне «залізо», за допомогою якого і будуть проводитися складні розрахунки. Причому вартість такого обладнання може бути дуже високою – захмарною. Займатися майнінгом має сенс тільки об'єднавшись в пули або створюючи обчислювальні ферми. по-друге, марність розрахунків, які

проводяться для майнингу.

Майнери постійно генерують нові блоки і споживають багато енергії. Однак розрахунки, які робить система, більше ніде не застосовуються. Вони лише забезпечують надійність мережі і не можуть бути використані в діловій сфері, в науці або будь-який інший галузі;

по-третє, атака 50%.

Вона також називається атакою більшості і полягає в тому, що користувач або група користувачів беруть під контроль переважну більшість видобувних потужностей. Відповідно, у них з'являється досить «сил» для контролю великого обсягу подій. Фактично, є можливість спробувати таку атаку з набагато меншим контролем мережі, але шанс на успіх буде дуже низьким. При атаці 50%, створення нових блоків монополізується. До того ж, такий користувач або група отримують винагороду і не дають іншим брати участь повноцінно в процесі.

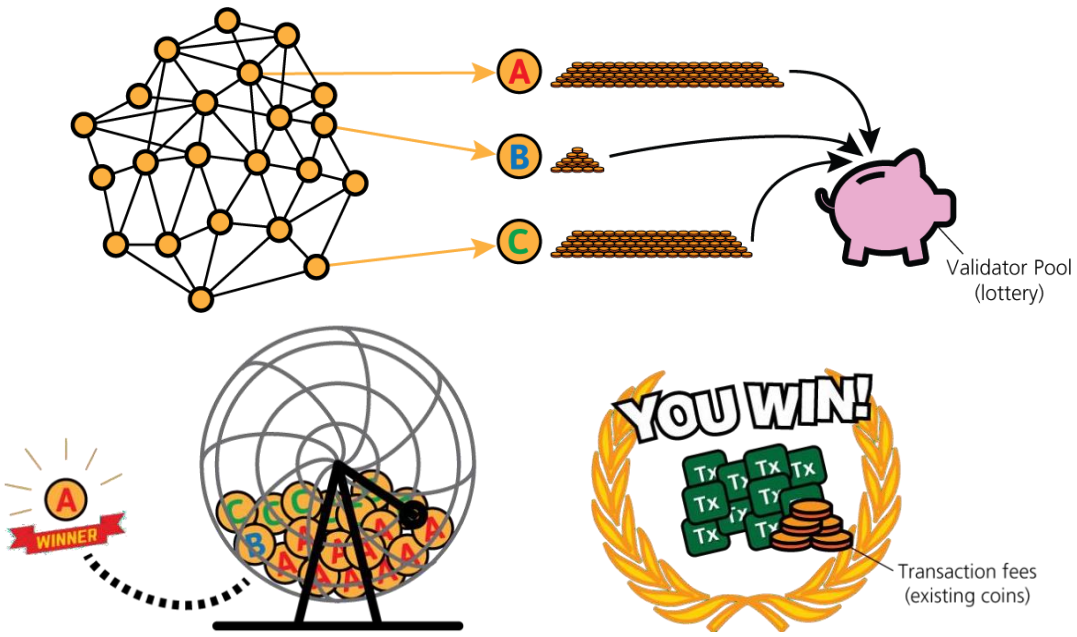
- Вчетверте, повільну швидкість роботи.

Ці недоліки (особливо недостатнья швидкість та значне споживання електроенергії для розрахунків) спонукали до створення нових протоколів консенсусу, заснованих на інших принципах.

1.3.2 Сутність консенсусного блокчейн-протоколу Proof-of-Stake

Proof-of-Stake (PoS) – перекладається як "доведення володіння часткою". Алгоритм працює за принципом: мережа довіряє валідатору, який володіє значною сумою у відповідній локальній валюті. Причому чим більша його частка (stake) у загальній сумі, тим вищими є його шанси на генерацію наступного блоку (й відповідно, отримання нагороди). У протоколі PoW нагороду отримують учасники, які вирішували криптографічні головоломки, щоб перевіряти транзакції та створювати нові блоки. У блокчейнах. що базуються на PoS (наприклад, Ethereum – майбутня реалізація Casper), вага голосів кожного валідатора залежить від розміру його депозиту (тобто

частки). Значні переваги PoS включають: порівняно високу швидкість та енергоефективність. Замість того, щоб конкурувати з іншими, майнери мережі закладають, ніби в ломбард, свої криптоактиви і чекають на випадкове обрання для валідації блокчейну.



Алгоритм обирає майнера для створення наступного блоку на основі внесеної частки власних монет. Відтак, якщо вузол має 5% від усіх часток у мережі в даний момент, він створить приблизно 5% блоків і перевірить 5% транзакцій. Ідея полягає в тому, що чим вище ставка валідатора у мережі, тим менше його зацікавленість у зловмисних діях.

Як і у випадку протоколу PoW, процедура вибору майнера для створення наступного блоку в PoS є ймовірнісною. Хоча транзакції обробляються відносно швидко в порівняні з Біткойном, PoS все ще вимагає використання токенів. Крім того, скептики вказують на те, що валідатори з високими ставками будуть вибиратися частіше і таким чином отримуватимуть ще більше токенів – багаті стають багатшими.

Загалом, кожен, хто має відповідну криптовалюту (у випадку Ethereum – ефір), може стати валідатором, відправивши спеціальний тип транзакції,

який передає ефір у депозит.

Переваги PoS на відміну від PoW:

- відсутня необхідність споживати велику кількість електроенергії, щоб забезпечити блокчейн;
- через відсутність високого споживання електроенергії, кількість монет, яка випускається для мотивації майнерів. може бути значно меншою;
- порівняно з PoW більша швидкодія;можливість використовувати економічні санкції, щоб зробити різноманітні форми атаки 51% набагато дорожчими для виконання, ніж при доведенні роботи.

Делегований Proof-of-Stake

Основний принцип роботи алгоритму Delegate-Proof-of-Stake (DPoS) – поділ голосуючих і валідуючих учасників. Учасники мережі, які мають право голосу в системі (власники монет) не є при цьому валідаторами транзакцій. З DPoS, власники монет використовують свої баланси для вибору списку вузлів, незалежно від їх балансу, що матимуть можливість брати участь у лотереї. Це стосується всіх власників монет, хоча це не винагороджує їх безпосередньо. Бюлетні вузлів з більш високим балансом матимуть більшу вагу на виборах. Власники також можуть голосувати за зміни параметрів мережі, що надає їм більший вплив та право власності на мережу.

Умови, в яких працює даний алгоритм консенсусу, відрізняються від умов, в яких працюють PoW і PoS. А саме, валідаторам необхідно розкрити свої особистості і заявити про готовність безперебійно підтримувати роботу повноцінного вузла мережі, своєчасно виконувати верифікацію транзакцій і формувати нові блоки.

Серед всіх користувачів проводиться голосування за кандидатів, де вага кожного голосу визначається сумою активів голосуючого. За результатами голосування вибирається N (зазвичай 20-50) кандидатів, які отримують право формувати нові блоки транзакцій. Правила протоколу гарантують коректне прийняття рішень, якщо більша частина активів, які беруть участь в голосуванні, контролюється чесними користувачами.

1.3.3 Опис атак на протоколи PoS та PoW

Зазвичай криптовалюти, що використовують PoS, мають меншу нагороду за сформований блок, оскільки ця система уникає дорогих обчислень. Це робить його більш екологічним, ніж PoW. Вартість "атаки 50%" на сучасний блокчейн, що використовує PoW, перевищує вартість атаки на блокчейн, що використовує PoS. У PoW зловмисник повинен буде придбати 50% обчислювальної потужності в мережі, але в PoS зловмисник повинен буде придбати 50% валюти в цій системі.

Хоча криптовалюта заснована не на довірі, а на стійкості криптографічних алгоримтмів, які використовуються для її побудови, це не означає, що в системі відсутні вразливості. Навпаки, звідси випливає, що зловмисники математично можуть експлуатувати ті чи інші "слабкі ланки" мережі. На протоколи консенсусу існують різноманітні атаки, розглянемо деякі з них.

DoS атака

Атака цього типу має на меті порушити нормальну роботу мережі крипто-валют шляхом заспамлення вузлів.

Sybil атака

Зловмисник створює ряд вузлів під різними іменами, коли жертва підключається тільки до контрольованих зловмисником вузлів.

Цензоршіп атака (вибіркове підтвердження блоків)

Ситуація, коли зловмисник вибірково підтверджує видобуті блоки, щоб досягти якоїсь своєї мети: змусити майнерів даремно витрачати ресурси, або відмінити транзакцію, або розділити мережу, тощо.

Атака розгалуження

Ця атака є набагато менш поширеною, ніж наступна (подвійної витрати). У літературі вона зустрічається всього у двух публікаціях. В англомовних джерелах вона називається "Splitting Attack". Вона полягає у

тому, що зловмисник намагається створити якомога довший форк, тобто створює та викладає свої блоки таким чином, щоб якнайдовше підтримувати існування двух ланцюжків однакової довжини. При цьому він не приховує самого факту побудови форку, і сторонній спостерігач може легко побачити, що більш довгий ланцюжок змінюється час від часу. Тобто деякі транзакції то зникають, то з'являються. Чесні майнери, які теж беруть участь у побудові блокчейну, повинні притримуватись протоколу майнінгу, тому змушені продовжувати то один, то інший ланцюжок, таким чином супроти свого бажання підтримуючи цей форк.

Робота, яка присвячена побудові оцінок стійкості блокчейна до цієї атаки, розглядає тільки випадок протоколу PoS. В цій роботі, при певних допущеннях, показано, що якщо частка стейку зловмисника $\frac{m}{m+n}$ строго менша за частку $\frac{n}{m+n}$ чесних майнерів, то імовірність форку експоненційно прямує до нуля, якщо довжина форку прямує до нескінченості. Іншими словами, для достатньо великих l імовірність форку довжини l має вигляд $e^{-f(l)}$, де $f(l)$ деяка додатня функція. При цьому отриманий результат є виключно асимптотичним, тому неможливо визначити, яка довжина l є "достатньо великою" для того, щоб можна було використати отримані оцінки. Відповідно, отримані результати не дають відповіді на питання: при заданих m, n та деякому (маленькому) $\varepsilon > 0$, якою має бути довжина l форку, щоб його імовірність була меншою за задане $\varepsilon > 0$.

Робота розглядає ту ж задачу, тобто ту ж саму модель атаки, що й, але для протоколу PoW. В цій роботі для моделей BitCoin та GHOST отримані наступні результати:

1) отримано аналітичні верхні оцінки імовірності форку заданої довжини l при заданих m, n та (маленькому) $\varepsilon > 0$;

2) по цим оцінкам отримані чисельні результати для різних часток зловмисника $\frac{m}{m+n}$.

Хоча отримані оцінки не виражені як експоненційні функції від довжини форку, але вони придатні для отримання чисельних результатів та побудови відповідних графиків. Причому графіки показують експоненційне спадання імовірності форку з ростом його довжини (при фіксованих m и n).

Ця атака є практично єдиною, для якої на даний час (в певних припущеннях) вдалось побудувати верхні оцінки імовірності успіху з урахуванням часу синхронізації мережі. Зокрема, у роботі було показано, що якщо зловмисник та чесні майнери мають приблизно однаковий час синхронізації, то імовірність форку є практично такою ж, як і для моделі з нульовим часом синхронізації; відмінності стають помітними лише при дуже значній долі зловмисника, близькій до 40-45%.

Зазначимо, що мета цієї атаки відразу не є очевидною , принаймні у порівнянні з наступною атакою (подвійної витрати). Однак ця атака має вагоме теоретичне значення у випадку, коли блокчейн використовується як основа для побудови криптографічних протоколів. З іншого боку, якщо блокчейн використовується в системі кріптовалюти, продовжуючи форк якомога довше, зловмисник хоче скомпрометувати мережу, щоб скупити максимальну кількість відповідної валюти за низькою ціною, розраховуючи на те, що через деякий час після компрометації ціна на цю криптовалюту знову підніметься. Але такий розрахунок є досить ефімерним. Зокрема, якщо криптовалюта базується на протоколі PoS, то для стейкхолдера вигідніше генерувати нові блоки та отримувати гарантований дохід, ніж марно ризикувати.

Якщо ж мова йде не про криптовалюту, а про блокчейн, який є розподіленою базою даних, то така атака має на меті створити невизначеність в базі даних та невпевненість серед її користувачів.

Атака подвійної витрати (Атака з підміною блоку)

Ця атака аналізується в багатьох публикаціях, серед основних можна назвати, а також в численній кількості популярних статей в інтернеті, і що важливо – у курсі Прингстонського университету "Bitcoin and

Cryptocurrency". В англомовних джерелах вона називається "Double Spending Attack". Її сутність полягає у тому, що зловмисник намагається використати свою монету як "нерозмінним п’ятаком" з добре відомої книги Аркадія та Бориса Стругацьких "Понеділок починається у суботу".

Технічно це відбувається так. Зловмисник у блоці з номером, наприклад, 5, виконує деяку транзакцію, в якій пересилає гроші за товар або послугу деякому постачальнику. Постачальник отримує ці гроші та надсилає товар покупцеві. Після отримання товару зловмисник намагається швидко створити інший, альтернативний блок з тим же номером 5, тобто блок, який посилається на блок з номером 4, але в якому ці ж самі гроші переводяться на іншу адресу – як оплата іншому постачальнику. або навіть собі в інший гаманець. Щоб гарантувати прийняття чесними майнерами його версії блокчейну, він напагається "причепити" на цей альтернативний блок з номером 5 якомога більше блоків. Якщо у нього вийде зробити альтернативний ланцюжок більшої довжини, то саме він, згідно з протоколом майнінгу, буде вважатись правильним. Очевидно, що чим більшою є частка зловмисника (не має значення, це частка обчислювальних потужностей, у випадку протоколу PoW, або частка стейку у випадку PoS), тим більше в нього шансів виконати цю атаку. Зокрема, якщо його частка більша за половину, то імовірність успіху цієї атаки дорівнює 1.

Оскільки ця атака є найпопулярнішою та найбільш реалізуємою на практиці, тому детально її розглянемо у наступному параграфі.

1.3.4 Атака з підміною блоку (Атака подвійної витрати). Порівняльний аналіз результатів різних авторів та основні недоліки цих результатів

Хоча назва цієї атаки вказує на її виникнення у галузі криптовалюти, технічно вона є атакою, напрвленою на підміну одного блоку у блокчейні іншим. Тому вона може застосовуватись до будь-якого блокчейну, незалежно від протоколу консенсусу, який лежить в його основі, та від способу

використання цього блокчейну – як для підтримки функціонування криптовалюти, так і для зберігання данних у децентралізованій розподіленій базі.

У літературі аналіз стійкості блокчейну до цієї атаки розглядався лише по відношенню до криптовалютних блокчейнів, але це аж ніяк не впливає на отримані щодо неї результати.

1.3.5 Результати Накамото та їх помилковість

Для захисту від атаки подвійної витрати, Накамото в своїй найпершій роботі [1] запропонував, щоб постачальник не надсилав товар відразу, як тільки транзакція з’явилася, а почекав деякий час, поки декілька блоків після цієї транзакції не будуть створені, і тільки потім, якщо транзакція не зникла з блокчейну, віддавати товар. У цьому випадку зловмисник не може будувати відкритий форк відразу після оплати, оскільки постачальник побачить, що транзакція то зникає, то з'являється в блокчейні, і відмовиться від угоди. На першому етапі атаки зловмисник спочатку чекає, поки після блоку з транзакцією з’явиться потрібна кількість блоків підтвердження. У цей час він може таємно генерувати альтернативний ланцюжок (форк), який відгалужується десь раніше, до блоку з транзакцією, тобто в термінах попереднього параграфу може генерувати альтернативний п'ятий блок і наступні за ним блоки, але у жодному разі не викладати цей альтернативний ланцюжок, щоб постачальник не здогадався про наміри зловмисника. Це є першим етапом атаки; перший етап продовжується до того моменту, коли постачальник надішле товар, після чого спробу зробити форк вже можна не приховувати. На другому етапі атаки, коли блоки підтвердження вже були сформовані і товар отримано, зловмисник або викладає свій альтернативний ланцюжок. якщо він довший від того, який побудували чесні майнери (тоді атака вже є успішною), або намагається "наздогнати" існуючий ланцюжок. Нехай, поки генеруються 6 блоків підтвердження, противник зміг знайти 4

блоки альтернативного ланцюжка. І тепер він відстає принаймні на 2 блоки. Якщо коли-небудь він зможе згенерувати таку кількість блоків, щоб "наздогнати" існуючий ланцюжок, який, у свою чергу, теж буде весь цей час зростати, то атака буде успішною. Зокрема, якщо йому вдалося згенерувати 7 або більше блоків на першому етапі атаки, поки він чекав потрібну кількість блоків підтвердження, то атака вдалася. Після отримання товару він просто викладає свій, більш довгий ланцюжок, в якому гроші залишаються у нього.

Зауважимо, що, на відміну від атаки з підміною блоку, атака розгалуження складається не з двох етапів, а з одного, причому весь час відбувається "явний" форк. Вигода зловмисника в атаці з підміною блоку очевидна: придбавши товар, він не витратив гроші. У випадку атаки розгалуження максимум, що він може зробити це скомпрометувати криптовалюту, що не несе йому ніякої вигоди, а якщо він теж володіє цією валютою, то навіть принесе збитки.

Тепер проаналізуємо результати різних авторів, в яких розглядається атака подвійної витрати і оцінюється стійкість блокчейну до цієї атаки. Зауважимо, що у всіх зазначених нижче роботах постановка задачі є наступною: при заданих m і n, а також маленькому $\varepsilon > 0$, потрібно знайти, якою має бути кількість блоків підтвердження z після транзакції, щоб ймовірність успішної атаки була менше заданого ε.

Як вже говорилося, перші результати по даному завданню були отримані в роботі Накамото. Однак отримані вони були в припущеннях, які не зовсім відповідають реальній моделі. Перше припущення, яке присутнє також майже у всіх інших роботах – це припущення про те, що час генерації блоку і час його появи в мережі збігаються. Тобто час затримки поширення блоку дорівнює нулю. Але з цього припущення випливає, що ймовірність "ненавмисного" форку дорівнює нулю, а в реальності такі форки трапляються приблизно 6 раз на місяць. Друге припущення є ще більш невдалим. Воно полягає в наступному: якщо ймовірність події дорівнює p, то кількість

випробувань, в яких відбудеться рівно n подій, дорівнюватиме $\frac{n}{p}$. Насправді це означає, що випадкову величину замінили її математичним сподіванням, що є, м'яко кажучи, некоректним.

У цих спрощених припущеннях отримані аналітичні вирази для ймовірності успіху атаки, за якими можна отримувати чисельні значення ймовірності. При цьому за отриманими виразами не є очевидним, що ймовірність успіху атаки зменшується експоненціально з ростом кількості блоків підтвердження, але побудовані згідно цим результатам графіки виглядають як спадні експоненціальні функції.

Далі проаналізуємо результати, наведені в роботі Накамото. Всі теореми, леми та припущення, що використовуються в цьому пункті, залишаємо в тому вигляді. в якому вони ввійшли до роботи.

Нехай зловмисник намагається виконати атаку та згенерувати більш довгий ланцюг блоків, ніж чесні майнери. Ймовірність того, що зловмисник коли-небудь наздожене альтернативний ланцюг за умови, що він відстає на z блоків від чесних майнерів, наведена у лемі 7.1.

Лема 1.1 Нехай p – ймовірність того, що чесний майнер створює наступний блок, q – ймовірність того, що наступний блок створено зловмисником. При цьому $p + q = 1$. Тоді ймовірність q_z того, що зловмисник коли-небудь наздожене альтернативний ланцюг за умови, що він відстає на z блоків дорівнює:

$$q_z = \begin{cases} 1, \ if_z q \geq p; \\ \left(\frac{q}{p}\right), \ else. \end{cases}$$

Доведення Леми 1.1. випливає з відомих результатів про так звану "задачу про банкрутство гравця".

Тобто випадку, коли $p > q$, ймовірність успіху атаки зменшується експоненціально з ростом числа блоків, на які відстає зловмисник.

Одержувач чекає, поки транзакція буде додана в блок і підтверджена z

блоками. Йому невідомо, скільки блоків зв цей час створив зловмисник. За (невірним) припущенням Накамото, кількість блоків зловмисника за цей час підпорядковується розподілу Пуассона з математичним сподіванням $\lambda = z\frac{q}{p}$.

Спробуємо пояснити, чому Накамото вибрав саме такий розподіл. Будемо вважати, що чесні майнери і зловмисники генерують блоки незалежно і «покроково», тобто на кожному наступному кроці був сгенерований один блок, і його згенерував чесний майнер з імовірністю p або зловмисник з імовірністю q. Тоді кількість кроків до того, як чесні майнери згенерують z блоків, має обернений біноміальний розподіл з математичним сподіванням $\frac{z}{p}$. Але тут Накамото наближає випадкову величину, що має обернений біноміальний розподіл, її математичним сподіванням, тобто він вважає, що кількість кроків до того, як чесні майнери згенерують z блоків, дорівнює точно $\frac{z}{p}$. Якщо прийняти таке наближення, то тоді імовірність того, що за цих $\frac{z}{p}$ кроків рівно k блоків згенерує зловмисник, дорівнює $C^k_{z/p} q^k p^{z/p-k}$.Далі Накамото застосовує наближення біноміального розподілу Пуасонівським: $C^k_{z/p} q^k p^{z/p-k} \approx \frac{e^{-\lambda}\lambda^k}{k!}$, де $\lambda = q \cdot \frac{z}{p}$, е незважаючи на те, що таке наближення допустиме лише для великих значень $\frac{z}{p}$ і маленьких значень q.

З використанням цього припущення імовірність успіху атаки визначається наступною теормою.

Теорема 1.1 Ймовірність того, що зловмисники коли-небудь наздоженуть альтернативний ланцюг за умови, що вони відстають на z блоків від чесних майнерів дорівнює:

$$q_z = \sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} 1, \ if \ k_{z-k} \geq z; \\ \left(\frac{q}{p}\right) \ , \ else. \end{cases}$$

При доведенні цієї теореми були зроблені суперечливі припущення: спочатку про те, що випадкові події "наступний блок згенеровано чесними майнерами" та "наступний блок згенеровано зловмисником" є незалежними, а потім про те, що відповідні їм бернулієвські випадкові величини є лінійно залежними, тобто їх сума дорівнює одиниці.

Як бачимо, результати Накамото ґрунтувалися на не зовсім вірних припущеннях, тому вони не є коректними.

1.3.6 Аналіз результатів Розенфельда, Пізнона та Зохара-Самполинського. Їх недоліки

В роботі Розенфельда були запропоновані інші, і як виявилося, більш точні аналітичні вирази для імовірності успіху атаки, при цьому для їх отримання була обрана трохи інша модель, ніж у Накамото. Однак в цій роботі не наводилося жодного обґрунтування обраної моделі. Автори припустили, що поява "чесних" / "нечесних" блоків в мережі описується негативним біноміальним розподілом, проте це припущення не було тут обґрунтовано. Водночас в цій роботі результати також були отримані в припущенні про те, що час поширення блоку в мережі дорівнює нулю. Незрозуміло, наскільки автори помітили помилковість другого допущення Накамото, проте вони не використовували його. Тому чисельні результати в цій статті [6] відрізняються від результатів Накамото, тобто для однієї і тієї ж ймовірності атаки в роботі Розенфельда потрібна більша кількість блоків підтвердження, що є природнім.

Основним результатом роботи є наступна теорема.

Теорема 1.2 Ймовірність успіху зловмисників після того, як z блоків були знайдені чесними майнерами, дорівнює:

$$P(z) = 1 - \sum_{k=0}^{n-1} \left(p^z q^k - q^z p^k \right) C_{k+z-1}^k .$$

В роботі Пінзона вперше було звернено увагу на невірність другого допущення Накамото. Точніше, там було сказано, що успіх зловмисника на першому етапі атаки буде істотно залежати від того часу, який знадобився на генерацію блоків підтвердження. До результатів Розенфельда теж було зауваження, що вони не зовсім точні, і була запропонована своя функція підрахунку ймовірності успіху на першому етапі. Проте формули, наведені у цій роботі, є незрозумілими ні з точки зору обгрунтування, ні з точки зору можливості їх використання для отримання чисельних виразів. Зазначимо, що ніяких чисельних результатів, отриманих за своїми формулами, автори не приводили.

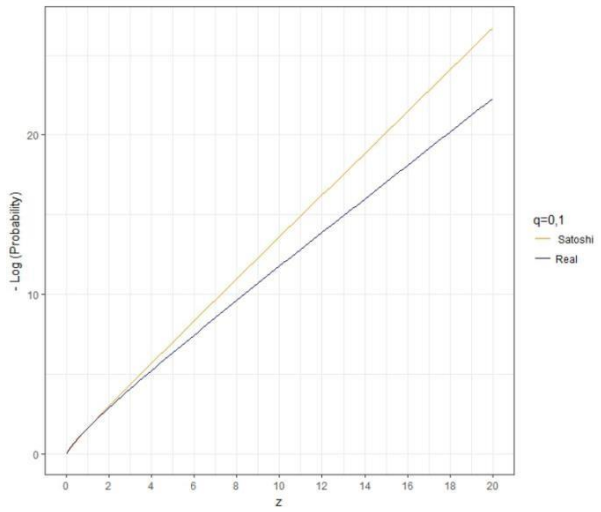
В кінці роботи говорилося, що результати робіт приблизно однакові, так як формули для ймовірності кількості блоків на першому етапі атаки приблизно рівні. Однак розрахунки показують, що чисельні результати Накамото і Розенфельда істотно відрізняються. Також слід зауважити, що в роботі Пінзона теж наявне припущення про миттєве поширення блоку в мережі.

На завершення цього параграфу зазначимо, вперше про те, що потрібно враховувати час синхронізації, було згадано в роботі. Проте далі автори не намагалися отримати результати в припущенні про ненульовий час синхронізації. У роботі доведені деякі твердження про співвідношення між швидкістю генерації блоку і швидкістю зростання основного ланцюжка, але одне з тверджень містить помилку, що руйнує весь хід доведення, а в інших твердженнях формули приведені без будь-якого пояснення. Підкреслимо, що в роботі немає жодного результату з аналітичними оцінками ймовірності атаки.

1.3.7 Результати досліджень Грунспан і Перес-Марко

Чудова, з математичної точки зору, робота Грунспан і Перес-Марко вражає математичною строгістю своїх тверджень і обґрунтувань. У цій роботі автори доводять те, що Розенфельд припустив без доведення: що процес генерації "чесних" / "нечесних" блоків в мережі описується негативним біноміальним розподілом. Однак автори не намагалися позбутися від припущення про миттєве поширення блоку в мережі. До речі, в цій роботі, з використанням спеціальних функцій, вперше доведено, що ймовірність форку спадає експоненціально з ростом його довжини.

Основний результат роботи Грунспана та Переса-Марко (у припущенні про нульовий час синхронізації) повністю співпадає з результатом Розенфельда, тобто теоремою 1.2. Проте в цій роботі він повністю обгрунтований. Зазначимо, що для обгрунтування результату було потрібно використати 6 видів імовірнісних розподілів: Бернуллі, біноміальний, негативний біноміальний, експоненційний, Ерланга та Пуасона. Тож зрозуміло, що його доведення є дуже нетривіальним, навіть з урахуванням припущення про нульовий час синхронізаціїї, який суттєво спрощує дослідження.



На осі x розташована кількість блоків z, на осі y розташоване значення логарифму (з мінусом) функції імовірності успіху зловмисників при обчислювальній потужності зловмисників $q = 0{,}1$. Жовтим кольором позначені чисельні значення функції ймовірності успіху зловмисників, що отримав Накамото, а синім – що отримав Грунспан.

Аналогічні результати представлені також у Таблицях 1.1. та 1.2.

Таблиця 1.1 - Ймовірності успіху згідно Накамото та Грунспана для $q = 0{,}1$.

z	$P(z)$	$P_{SN}(z)$
0	1,0000000	1,0000000
1	0,2000000	0,2045873
2	0,0560000	0,0509779
3	0,0171200	0,0131722
4	0,0054560	0,0034552
5	0,0017818	0,0009137
6	0,0005914	0,0002428
7	0,0001986	0,0000647
8	0,0000673	0,0000173
9	0,0000229	0,0000046

Таблиця 1.2 Ймовірності успіху згідно Накамото та Грунспана для $q = 0{,}3$.

z	$P(z)$	$P_{SN}(z)$
1	2	3
0	1,0000000	1,0000000
5	0,1976173	0,1773523
10	0,0651067	0,0416605
15	0,0233077	0,0101008
20	0,0086739	0,0024804
25	0,0033027	0,0006132
30	0,0012769	0,0001522

1	2	3
35	0,0004991	0,0000379
40	0,0001967	0,0000095
45	0,0000780	0,0000024

За Таблицями 1.1 та 1.2 видно, що результати, наведені Грунспаном, є "гіршими" у порівнянні з тими, що були наведені Накамото, тобто імовірність атаки є більшою. Це пояснюється некоректністю припущень, зроблених Накамото.

1.4. Атака з підміною блоку для протоколу консенсусу PoS. Імовірність успіху атаки в залежності від кількості блоків підтвердження

Метою та способом виконання атака з підміною блоку для випадку протоколу PoS не відрізняється від однойменної атаки для протоколу PoW. У цьому розділі знайдемо явні вирази для імовірності успіху атаки з підміною блоку у випадку протоколу PoS і, зокрема, покажемо, що ця імовірність залежить від кількості блоків підтвердження та частки зловмисника у мережі.

Дуже важливим є той факт, що для протоколу PoS час синхронізації мережі практично не має значення, оскільки таймслот, протягом якого стейкхолдер має створити блок, як правило, є набагато більшим за час синхронізації. Крім того, час виходу блоку для протоколу PoS суттєво менший, ніж для протоколу PoW, оскільки кількість виконаної роботи є суттєво меншою. Ці два фактори забезпечують значну перевагу протоколу PoS.

Для подальшого викладення введемо деякі позначення. Нехай $B_0, B_1, ..., B_n$ – блоки, які входять до "правильної" версії блокчейну, тобто є створеними чесними майнерами. Транзакція X, в якій зловмисник перерахував кошти постачальнику, була включена майнрами у блок деякий блок B_i, $i \in N$. За правилами прийняття транзакції, постачальник чекає z

блоків підтвердження після блоку B_i. Як було зазначено раніше, це робиться для того, щоб зловмиснику було складно підмінити ланцюжок, який містить блок B_i, іншим ланцюжком, який містить блок з альтернативною транзакцією Y. Задача полягає у визначенні кількості блоків підтвердження, які, при заданій частці зловмисника, гарантують, що імовірність успіху атаки з підміною блоку не перевищує деяке задане значення, наприклад, 10^{-3}.

Для визначеності припустимо, що зловмисник будує форк, починаючи з блоку B_{i-1}, який передує блоку з транзакцією X. У цьому параграфі розглянемо дві стратегії виконання атаки та обчислимо імовірність успіху кожної з них.

1.4.1. Стратегія 1: зловмисник не підтверджує блоки з основної гілки

Противник не формує блоки в основному ланцюжку під час своїх таймслотів. Ці таймслоти він використовує для формування альтерналивного ланцюжка, який починається з блоку B_{i-1}. але після того, як z блоків $B_{i+1}, B_{i+2}, ..., B_{i+z}$ сформовані, він намагається створити альтернативну гілку, починаючи з блоку, який передує блоку B_i. Зазначимо, що ця гілка повинна обов'язково починатись до блоку B_i, інакше блок з транзакцією Y буде містити некоректну транзакцію, яка використовує вже витрачені монети, і буде вилучений з блокчейну разом з усіма блоками, що на нього посилаються.

Нехай альтернативним ланцюжком з точкою розгалуження у точці B_{i-1} є ланцюжок $B_1, ..., B_{i-1}, B_i', B_{i+1}', ...$, де B_i' – блоки, сформовані зловмисником. Важливо, що відповідно до цієї стратегії (противник не формує свої блоки в "чесних" ланцюжках) всі блоки в ланцюжку $B_{i+1}, B_{i+2}, ..., B_{i+z}$, а також блоки B_{i-1} та B_i, сформовані чесними учасниками. Щоб досягти успіху, противник повинен побудувати альтернативний ланцюжок, який довший, ніж "чесний". Це можливо лише тоді, коли в для деякого s після сформованих блоків

$B_{i+1}, B_{i+2}, ..., B_{i+z}$, кількість таймслотів противника між слотом $t(B_{i-1})$ та слотом з номером s не менше, ніж кількість "чесних" слотів за один і той же інтервал часу. У цьому випадку він може утворити ланцюг:

$$B_0, ..., B_{i-1}, B_i', B_{i+1}', ..., B_r'$$

для деякого r, де всі блоки $B_i', B_{i+1}', ..., B_r'$ створені у тих часових інтервалах, що належать зловмиснику, а B_r' утворюється в часовому інтервалі під номером s.

Отже, необхідною і достатньою умовою успішної атаки є наявність такої послідовності часових інтервалів після $t(B_{i-1})$, коли кількість слотів, що належать зловмиснику, не менша за кількість "чесних" слотів.

Припустимо, що серед n учасників рівно t $(t < n/2)$ є зловмисниками і $n-t$ чесними. Отже, $p = (n-t)/n$ – це імовірність того, що наступний таймслот належить чесному майнеру, а $q = t/n$ – імовірність альтернативної події.

Нехай ξ_i, $i \geq 1$ – послідовність випадкових величин, що приймають два значення:

$$\xi_i = \begin{cases} -1, \text{ with probability } q, \\ 1, \text{ with probability } p. \end{cases}$$

де -1 відповідає таймслотам зловмисника, 1 відповідає таймслотам чесних майнерів.

Визначимо наступні випадкові величини:

$$S_0 = 0, S_n = \sum_{i=1}^{n} \xi_i;$$

$$S_0^- = 0, S_n^- = \sum_{i=1}^{n} (-\xi_i \vee 0) \text{ та } S_0^+ = 0, \quad S_n^+ = \sum_{i=1}^{n} (\xi_i \vee 0). \tag{1.3}$$

Для деякого $k \in N$ визначимо ще одну випадкову величину:

$$\tau_k = min\{l \geq 1 : S_l^+ = k\}.$$

Тепер задачу про обчислення імовірності успіху атаки може бути сформулювати як задача обчислення імовірності наступної події для $k = z+1$

:

$$A(k) = \{\exists\, m > \tau_k : S_m^- \geq S_m^+\},$$

Для подальших визначень знадобиться результат , що стосується випадкових блукань, а саме лема про розорення гравця [9].

В позначеннях (1.1)-(1.3) визначимо випадкові величини:

$$S_n^{(k)} = S_n + k\,,\ S_0^{(k)} = k\,.$$

Також визначимо подію $C_k = \left\{ \exists\, l \in \mathrm{N} : S_l^{(k)} = 0 \right\}$ і її імовірність позначимо $q_k = P\left(C_k\right)$. Тоді за Лемою 1.1:

$$q_k = \begin{cases} 1, & if_k\ q \geq p, \\ \left(\dfrac{q}{p}\right), & else. \end{cases}$$

Щоб довести основний результат про імовірність успіху атаки з підміною блоку, нам потрібні деякі визначення та властивості спеціальних функцій.

Означення 1.1. Регулярною неповною бета-функцією називається функція

$$I_x(a,b) = \sum_{l=a}^{\infty} C_{b+l-1}^{l}\, x^l (1-x)^b = \frac{B_x(a,b)}{B(a,b)},$$

де $B_x(a,b) = \int_0^x t^{a-1}(1-t)^{b-1}\,dt$ – неповна бета функція,

$B(a,b) = B_1(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}\,dt = \dfrac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ – бета функція,

$\Gamma(x) = \int_0^{\infty} t^{x-1}e^{-t}dt$ – гама функція.

Лема 1.2: Регулярна неповна бета-функція задовольняє співвідношенню симетрії:

$$I_p(a,b) + I_q(a,b) = 1, \text{ для } 0 \leq p,q \leq 1,\ p+q = 1.$$

З Леми 1.1 та визначення від'ємного біноміального розподілу отримуємо наступний наслідок.

*Наслідок 1.1:*В заданих позначеннях: $\sum_{l=0}^{z} C_{z+l}^{l} q^{z+1} p^{l} = \sum_{l=z+1}^{\infty} C_{z+l}^{l} p^{z+1} q^{l}$.

Тепер сформуємо основний результат цього параграфу.

*Теорема 1.3:*У заданих позначеннях імовірність P_z успіху атаки з підміною блоку за умови, що отримано z блоків підтвердження, дорівнює:

$$P_z = \begin{cases} 1, & \text{if} \quad q \geq p; \\ P(A(z+1)) = 2\sum_{l=0}^{z} C_{z+l}^{l} p^{l} q^{z+1}, & \text{else.} \end{cases}, \tag{1.5}$$

або, використовуючи локальну теорему Муавра-Лапласа, для відповідних p, q, та z:

$$P_z = 2p \sum_{l=0}^{z} \frac{\varphi\left(\frac{zq - lp}{\sqrt{(z+l)pq}} \right)}{\sqrt{(z+l)pq}},$$

або, використовуючи регулярну неповну бета функцію:

$$P_z = 2I_q(z+1, z+1),$$

що для досить великих z можна записати як

$$P_z = \mathrm{O}\left((4pq)^{z+1} \right).$$

Доведення. Визначимо наступні події:

$$H_l = \{\tau_{z+1} = z+1+l\} = \{S^{-}_{\tau_{z+1}} = l\}, \quad l \in \{0,1,...\},$$

де H_l – подія, яка означає, що супротивник накопичив l блоків до того часу, коли почався слот з номером τ_z. Важливо, що події H_l, $l \in \{0,1,...\}$, утворюють повну групу подій.

Тоді за формулою повної ймовірності:

$$P(A(z+1)) = \sum_{l=0}^{\infty} P(A(z+1) / H_l) P(H_l).$$

Імовірність події H_l, $l \in \{0,1,...\}$, визначається як

$$P(H_l) = C^l_{z+1+l-1} p^{z+1} q^l = C^l_{z+l} p^{z+1} q^l ,$$

де

$$\sum_{l=0}^{\infty} C^l_{z+l} p^{z+1} q^l = 1.$$

Відповідно до Леми 1.1,

$$P(A(z+1) / H_l) = \begin{cases} \left(\frac{q}{p}\right)^{z+1-l}, & \text{if } q < p \text{ and } l < z+1; \\ 1, & \text{else.} \end{cases}$$

Перепишемо (1.9), використовуючи (1.10)–(1.12):

$$P(A(z+1)) = \sum_{l=0}^{z} C^l_{z+l} p^{z+1} q^l \left(\frac{q}{p}\right)^{z+1-l} + \sum_{l=z+1}^{\infty} C^l_{z+l} p^{z+1} q^l =$$

$$= \sum_{l=0}^{z} C^l_{z+l} p^{z+1} q^l + \sum_{l=z+1}^{\infty} C^l_{z+l} q^{z+1} p^l =$$

$$= 1 - \sum_{l=z+1}^{\infty} C^l_{z+l} p^{z+1} q^l + \sum_{l=z+1}^{\infty} C^l_{z+l} q^{z+1} p^l .$$

З означення 1, формули (1.4) та леми 1.1, а також наслідку 1.1 і формули (1.13) отримаємо

$$P\left(A(z+1)\right) = 1 - I_p(z+1, z+1) + I_q(z+1, z+1) = 2I_q(z+1, z+1) = 2\sum_{l=0}^{z} C^l_{z+l} q^{z+1} p^l ,$$

і формули (1.5) і (1.7) доведені.

Щоб довести формулу (1.6), для відповідних z, p та q (якщо $z \cdot p \cdot q > 25$ або $p \le 0{,}9$ та $n \cdot p \cdot q > 5$) останній вираз можна записати як $C^l_{z+l} p^{z+1} q^l$ або $pC^l_{z+l} p^z q^l$ та застосувати локальну теорему Муавра-Лапласа:

$$C^l_{z+l} p^z q^l = \frac{\varphi\left(\frac{zq - lp}{\sqrt{(z+l)pq}}\right)}{\sqrt{(z+l)pq}},$$

де $\varphi(x)$ – стандартна нормальна щільність розподілу, $\varphi(x) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}$.

Для доведення формули (1.8) зауважимо, що

$$I_q(z+1,z+1) = I_{4q(1-q)}\left(z+1,\frac{1}{2}\right) = I_{4qp}\left(z+1,\frac{1}{2}\right), \text{ коли } 0 \le q \le \frac{1}{2}.$$

Для фіксованих x, b ($b>0$, $0<x<1$), при $a \to \infty$, для кожного $n = 0,1,...$ наступна рівність вірна:

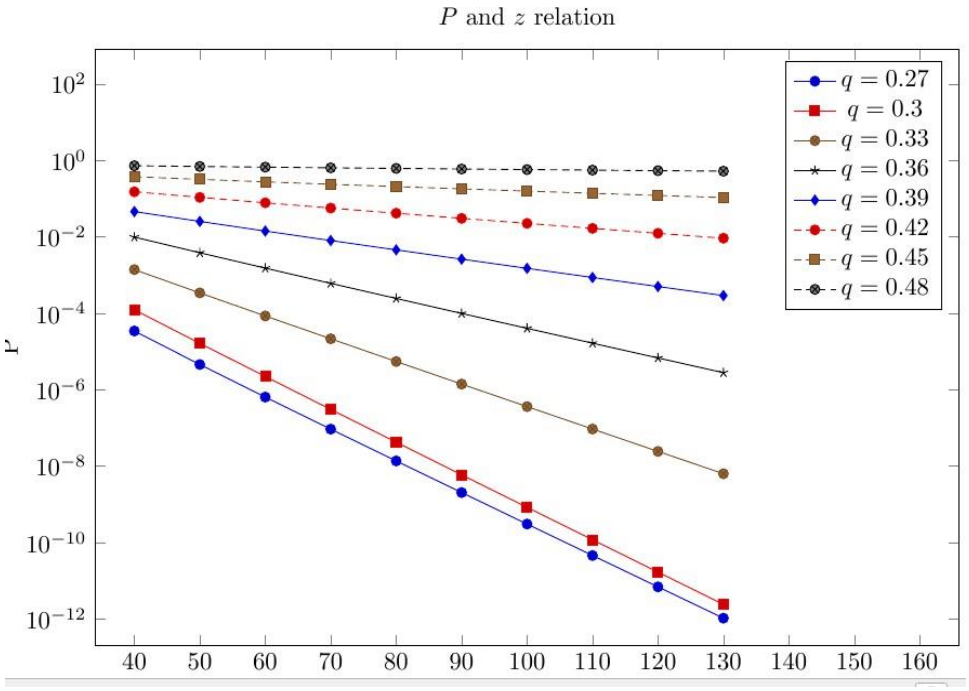
$$I_x(a,b) = \Gamma(a+b)\, x^a (1-x)^{b-1} \times$$
$$\times\left(\sum_{k=0}^{n-1} \frac{1}{\Gamma(a+k+1)\Gamma(b-k)}\left(\frac{x}{1-x}\right)^k + O\left(\frac{1}{\Gamma(a+n+1)}\right)\right).$$

Отже, для $n=0$ отримуємо:

$$P(A(z+1)) = 2I_q(z+1,z+1) = I_{4pq}\left(z+1,\frac{1}{2}\right) =$$
$$= \Gamma(z+1.5)(4pq)^{z+1}(1-4pq)^{-\frac{1}{2}} \times O\left(\frac{1}{\Gamma(z+2)}\right) = O\left((4pq)^{z+1}\right).$$

Теорема доведена.

На Рисунку 1.6 нижче приведена залежність значення логарифма ймовірності форку P_z, визначеній в формулі (1.7) (по осі Y), від значення z (по осі X), для різних часток зловмисника. Оскільки графіки для логарифма ймовірності є прямими лініями, то саме значення P_z спадає експоненційно з ростом z. Згідно формули (1.8), швидкість спадання функції $P(A(z+1))$ з ростом z така сама, як у функції $(4pq)^{z+1}$.



У Таблиці 1.1. наведено мінімальні значення z для різних значень частки зловмисника q, які забезпечують умову $P\left(A(z)\right)<10^{-3}$.

Таблиця 7.1 Мінімальна кількість блоків підтвердження, при якій $P_z<10^{-3}$

q	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45
z	10	10	15	20	25	60	150	540

1.4.2. Стратегія 2: зловмисник підтверджує блоки з основної гілки

У цій стратегії зловмисник, видаючи себе за чесного учасника, формує свої блоки на основному ланцюжку до того моменту, коли транзакція отримає необхідну кількість підтверджен. Після цього починає атаку шляхом формування альтернативного ланцюжка.

Як і в попередньому випадку, в альтернативному ланцюжці він може використовувати всі свої часові проміжки після того моменту, коли блок B_{i-1} був сформований. Основна відмінність полягає в тому, що всі блоки цього

ланцюга утворюються в послідовних таймслотах без пропусків, тобто для створення z блоків підтвердження для блоку B_i потрібно лише z часових інтервалів.

Для створення альтернативного ланцюжка, починаючи з блоку B_{i-1}, зловмисник може використовувати всі власні таймслоти (після слоту з номером $i-1$, коли B_{i-1} був сформований).

Будемо користуватись введеними раніше позначеннями. Для деякого $k \in \mathrm{N}$ визначимо наступну подію:

$$E(k) = \{\exists\, m \geq k : S_m^- \geq S_m^+\}.$$

Тоді для обчислення імовірності успіху атаки достатньо обчислити $P(E(z+1))$.

Теорема 1.4: у заданих позначеннях виконується рівність

$$P(E(z+1)) = (2q)^{z+1}.$$

Доведення. Визначимо події:

$$H_l = \{S_{z+1}^- = l\}, l = \overline{0, z+1}.$$

Подія H_l полягає у тому, що противник накопичив l таймслотів між $t(B_{i-1})$ та $t(B_{i+z})$. Крім того, події $H_l, l \in \{0,1,...\}$ утворюють повну групу подій.

Тоді за формулою повної імовірності:

$$P(E(z+1)) = \sum_{l=0}^{z+1} P(E(z+1) / H_l) P(H_l).$$

Імовірності подій H_l, $l \in \{0,1,...\}$ визначаються як імовірностю біноміального розподілу:

$$P(H_l) = C_{z+1}^{l}\, q^l\, p^{z+1-l}, \quad l = \overline{0, z+1}.$$

Імовірності $P(E(z+1) / H_l)$ отримаємо з використанням Леми 1.1:

$$P(E(z+1) / H_l) = \begin{cases} \left(\frac{q}{p}\right)^{z+1-l}, & if\ \ q < p\ \ and\ \ l \leq z+1; \\ 1, & else. \end{cases}$$

Перепишемо (1.15), використовуючи (1.16) та (1.17):

$$P(E(z+1)) = \sum_{l=0}^{z+1} C_{z+1}^{l} q^{l} p^{z+1-l} \left(\frac{q}{p}\right)^{z+1-l} =$$

$$= \sum_{l=0}^{z+1} C_{z+1}^{l} q^{l} q^{z+1} = q^{z+1} \sum_{l=0}^{z+1} C_{z+1}^{l} = q^{z+1} \cdot 2^{z+1} = (2q)^{z+1}.$$

Теорема доведена.

Під час порівняння формул (1.7) та (1.14) можна побачити, що для противника вигіднішою є перша стратегія. Дійсно, за умови $p > \frac{1}{2} > q$ отримаємо наступну нерівність:

$$4pq > 4 \cdot \frac{1}{2} q = 2q,$$

отже, імовірність форку в другій стратегії менша за його імовірність в першій стратегії, при тих же значеннях q та z.

1.5. Модифікація атаки з підміною блоку: атака з розділенням потужностей

В цьому пункті опишемо модифікацію атаки з підміною блоку, яку назвемо "атака з розділенням потужностей", та обчислимо верхню оцінку імовірності її успіху.

Основна відмінна риса цієї модифікації полягає в тому, що зловмисник намагається зменшити обчислювальний ресурс чесних майнерів, створюючи такі обставини, що частина цього ресурсу витрачається даремно. В класичній атаці зловмисник впродовж всієї атаки генерує свій ланцюжок таємно, аж до того моменту, як він стане довшим за основний. Коли цей момент настає, він оприлюднює цей згенерований ланцюжок і чесні майнери переключаються на нього. Натомість в запропонованому варіанті атаки зловмисник оприлюднює ланцюжок в той момент, коли його довжина достягає довжини чесного ланцюжка, після цього атака буде успішною лише в тому випадку,

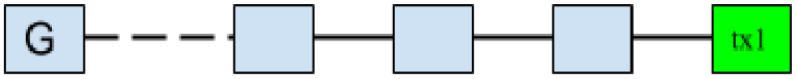
коли виконуються наступні умови:

1) чесні майнери розділяться на дві гілки та одночасно будуть генерувати два ланцюжки;

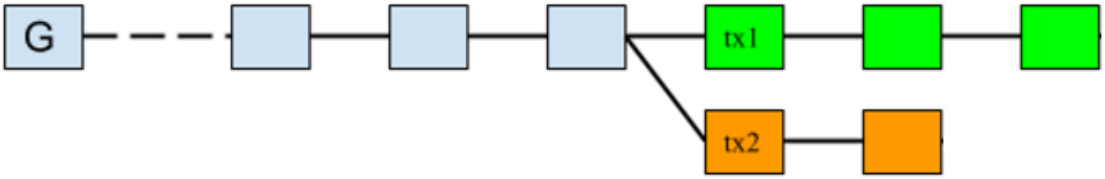
2) першим буде згенерований блок на чесній гілці. Інакше про існування альтернативної гілки, а отже і про атаку, стане відомо до її успішного завершення.

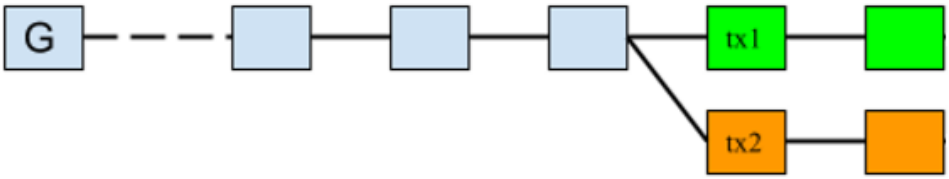
Опис атаки:

Зловмисник *A* хоче купувати товари у постачальника *B*. Для цього *A* створює транзакцію tx_1 з оплатою *B* та надсилає її до блокчейну.



На наступному кроці зловмисник негайно починає генерувати альтернативну (помаранчеву) гілку блокчейну, але не оприлюднює її. Він повинен завжди підтримувати стан системи, в якому ланцюжок чесних майнерів повинен бути більшої або такої ж самої довжини, як і альтернативний ланцюжок.





В випадку, якщо першим буде оприлюднений блок альтернативної гілки, атака не відбудеться, оскільки в цьому випадку чесні майнери переключаться на генерацію альтернативної гілки і це призведе до того, що продавець отримає сигнал, що його транзакція не отримала достатньо підтверджень і просто зникла з блокчейну.

Поки ці умови виконуються, продавець не отримує сигналу про існування альтернативної гілки. Сигнал піде тільки тоді, коли альтернативний буде обганяти чесни ланцюжок. А доти продавець не бачить альтернативної транзакції, та не розуміє що на нього проводиться атака і просто надалі чекає необхідну кількість блоків підтвердження.

Після того, як продавець відправив зловмиснику товар, зловмисник може відкрити альтернативну гілку, якщо вона довша за основну. Тоді транзакція tx_1 буде вилучена з блокчейну. Замість неї буде включена транзакція tx_2. Блокчейн буде продовжуватись з ланцюгом зловмисника, тому оплата продавцю *B* буде втрачена назавжди. В той же час зловмисник *A* отримає і товари, і гроші.

Тепер розглянемо, чому атака має саме таку назву. Коли існують одночасно дві гілки однакової довжини, то чесний майнер повинен вибрати ту, в якій "кількість роботи" є більшою. Але, якщо враховувати, що час синхронізації не є нульовим, можливою є ситуація, коли частина майнерів бачать одну з цих гілок довшою. Якщо вважати, що кожен майнер з рівною імовірністю почне продовжувати ту чи іншу гілку, то сумарна протужність всіх чесних майнерів розділиться навпіл. При появі кожного нового блоку

імовірність такої ситуації дорівнює $\frac{1}{2}$,ому верхня оцінка імовірності такої атаки за умови наявності z блоків підтвердження дорівнює $\left(\frac{1}{2}\right)^z$, тобто ця

імовірність спадає експоненційно з ростом кількості блоків підтвердження.

Підсумки розділу

У цьому розділі було наведено основні принципи формування блокчейну, основні протоколи консенсусу – Proof-of-Work та Proof-of-Stake, які він може використовувати, та основні типи загроз для цих протоколів. Найбільше уваги було приділено двом атакам: атаці розгалуження та атаці з підміною блоку, яка у випадку криптовалютного блокчейну отримала назву "атака подвійної витрати".

Щодо атаки з підміною блоку на блокчейн з протоколом PoW, то були наведені результати різних авторів щодо імовірності цієї атаки, починаючи з Накамото, та показано, що отримані ними результати або є некоректними, або отримані у такій моделі, що є далекою від реальної. Питання про імовірність цієї атаки для протоколу консенсусу PoW в рамках моделі, яка є близькою до реальної ситуації, на даний час є відкритим. Отримання оцінок для цієї імовірності може бути темою подальших досліджень.

Що стосується атаки з підміною блоку на блокчейн з протоколом PoS, то у цьому розділі отримані точні аналітичні вирази для імовірності цієї атаки, а також відповідні чисельні приклади та графіки.

Для обох протоколів показано, що верхні асимптотичні оцінки імовірності атаки на підміну блоку спадають експоненційно з ростом кількості блоків підтвердження.

2. ШЛЯХИ УЗАГАЛЬНЕННЯ І УДОСКОНАЛЕННЯ БЛОКЧЕЙН ТЕХНОЛОГІЇ З ВИКОРИСТАННЯМ РІЗНИХ ПРОТОКОЛІВ КОНСЕНСУСУ

Сучасні криптовалюти забезпечують циркуляцію децентралізованих платежів, з використанням загальнодоступного журналу транзакцій (transaction ledger), без залучення довіреної сторони. Журнал транзакцій є результатом загальної праці всіх учасників децентралізованої мережі. Він створюється на базі протоколу консенсусу, який є захищеним від дій зловмисника (нечесних майнерів) з обмеженим ресурсом.

Проте протокол, запропонований Накамото, який використовує консенсус PoW з параметрами, встановленими у мережі Bitcoin, має низьку пропускну здатність. Середня швидкість обробки транзакцій у мережі Bitcoin – всього близько 7 транзакцій за секунду, тобто менше ніж 500 000 транзакцій за день. Для деяких інших децентралізованих криптовалют ці значення є трохи більшими, але все одно залишаються дуже малими у порівнянні з централізованим фінансовим обслуговуванням. Такі обмеження для консенсусу, запропонованого Накамото, визначаються обмеженнями на розмір блоку та на інтенсивність генерації блоків. Безпосередня зміна цих параметрів відразу призводить до таких небажаних явищ, як часткова централізація або суттєве зниження стійкості до атаки подвійної витрати або атаки розгалуження. Наприклад, збільшення розміру блоку призводить до збільшення часу затримки блоку, що суттєво підвищує імовірності атаки подвійної витрати та атаки розгалуження. Зменшення часу виходу блока (а, отже, і зменшення кількості роботи для створення блоку) призводить до різних негативних явищ, а саме: до збільшення імовірності ненавмисного форку, а отже і до втрати великої кількості "роботи"; до збільшення імовірності атак – за умови наявності добре синхронізованого зловмисника. На даний час проблема форку вирішується, згідно протоколу Накамото, за правилом більш довгого ланцюжка. Однак при зміні одного з названих

параметрів такий спосіб консенсусу уже, як бачимо, не захищає мережу від різних вразливостей та проблем. Це означає, що при збільшенні розміру блоку або інтенсивності виходу блоків частина обчислювальних потужностей чесних майнерів ніби "втрачається", оскільки чесні майнери чітко виконують децентралізований протокол консенсусу, а відносна доля обчислювальних потужностей зловмисника зростає, в тому числі – за рахунок використання переваги кращої синхронізації своїх керованих обчислювальних потужностей. Тоді, враховуючи, що час розповсюдження інформації про сгенерований блок є скінченою, виникає ситуація, коли для того, щоб імовірність успішної атаки була рівна 1, достатньо щоб у зловмисника було не 50% потужностей, а суттєво менше. Наприклад, у мережі з достатньо великим часом затримки сигналу про верифіковані блоки та високою інтенсивністю виходу блоків атака подвійної витрати можлива вже при наявності 30% хешрейту зловмисника. Тому питання про створення протоколу консенсусу, який би дозволив б більш швидку обробку транзакцій зі збереженням стійкості до основних атак, є дуже актуальним. Та компанія, яка зможе досягти цього і створити криптовалюту з потрібними параметрами, буде мати величезну перевагу перед своїми конкурентами. Тому тема масштабування блокчейну зі збереженням його корисних властивостей активно обговорюється та розвивається.

2.1. Огляд основних проблем, для вирішення яких необхідне масштабування блокчейну

На сьогоднішній день жоден зі способів масштабування блокчейну, запропонованих у різних роботах, не є досконалим. В одних знайдені суттєві вразливості до атак; інші пропонують лише частковий розв’язок зазначених проблем. Після ретельного дослідження величезної кількості всіх таких пропозицій навіть виникає підозра, що неможливо так змасштабувати блокчейн, щоб результуючий протокол консенсусу на блокграфі мав,

принаймні частково, всі бажані властивості.

Спочатку назвемо всі проблеми, які існують сьогодні у мережі блокчейн та які бажали вирішити дослідники у запропонованих ними протоколах консенсусу на блокграфі.

1. Швидка обробка транзакцій. Може досягатись як за рахунок збільшення об'єму блоків, так і за рахунок збільшення інтенсивності їх виходу.

2. Збільшення реальної децентралізації. Єдиний шлях вирішення – зменшення об'ємів ресурсу, необхідного для створення блоку. Під ресурсом розуміється або обчислювальні потужності (для протоколу PoW), або депозит у відповідній криптовалюті (для протоколу PoS), або інший аналогічний ресурс, в залежності від протоколу консенсусу. На сьогоднішній день децентралізація блокчейну є лише відносною, оскільки необхідність мати досить великий ресурс для генерації блоків примушує майнерів об'єднуватись у величезні майнінгові пули.

3. Зменшення часу очікування підтвердження транзакції. Наприклад, у мережі Bitcoin транзакція вважається підтвердженою, якщо після блоку, що її містить, вийшло не менше шести блоків. Ці блоки називаються "блоки підтвердження". Оскільки в середньому блок виходить 1 раз на 10 хвилин, то підтвердження транзакції доводиться очікувати в середньому 1 годину. І це тільки середній час очікування; реальний час може суттєво відрізнятись завдяки великій дисперсії.

4. Лінійний порядок блоків. При переході до графчейну доводиться багато зусиль витратити на властивість, яка дістається блокчейну "задарма" – збереження лінійного порядку блоків. Ця властивість, яка наче б то не відіграє значної ролі для обробки транзакцій, виявляється надзвичайно важливою у протоколах, що працюють зі смарт контрактами (наприклад, у блокчейні криптовалюти DASH). В цих випадках існування лінійного порядку блоків необхідно для коректності процедури голосування за контракти.

5. Збереження стійкості блокграфу до основних типів атак. Для блокчейну існують різні способи доведення стійкості до основних атак: атаки подвійної витрати, атаки розгалуження, цензоршіп атаки (відмови у обробці транзакції або блока), атаки на зміну лінійного порядку (для блокчейну це просто частковий випадок атаки подвійної витрати). Для деяких атак, наприклад до атаки подвійної витрати, отримано аналітичні верхні оцінки імовірності у самих загальних припущеннях; до деяких інших атак існують лише асимптотичні оцінки у припущенні, що час є неперервною величиною. Побудова лінійного порядку на блокграфі, який буде стійким до атак, є надзвичайно трудомісткою задачею.

2.2. Огляд основних протоколів консенсусу на блокграфах

Термін "блокграф" найбільш точно відображає ту конструкцію, якою дослідники намагаються узагальнити традиційний блокчейн. На сьогодні найбільш вживаною є абревіатура DAG – орієнтований ациклічний граф. Хоча у різних публікаціях автори використовують різні назви: графчейн, гешграф, DAG (direted asyclic graph), інші. Кожна назва відображає якісь свої особливості запропонованого блокграфу. Надалі будемо вживати цей термін як синонім до терміну "графчейн".

Однією з найперших робіт, у якій було наведено узагальнення блокчейну, є робота, в якій було запропоновано структуру під назвою GHOST. В цій роботі замість лінійної структури, введеної Накамото, було побудовано іншу, більш загальну структуру, що являла собою "дерево" блоків. На відміну від протоколу Накамото, коли кожен блок має лише одного "нащадка", який на нього посилається, структура GHOST допускає існування, взагалі кажучи, довільної кількості таких нащадків. Іншими словами, ця структура допускає існування форків. Форки можуть виникати з двох причин: поганої синхронізації мережі або під час атаки зловмисника. Два блока, побудованих на одній "висоті", можуть обидва бути валідними.

Кожен лінійний ланцюжок цього дерева, який має несуперечливу історію транзакцій, є валідним.

Такий підхід до консенсусу дозволяє збільшити пропускну здатність протоколу, хоча все одно деяка кількість створених блоків може "втратитись".

Згідно протоколу, кожен майнер, при наявності кількох ланцюжків, може обирати не той, який йому видається більш "правильним". У випадку виникнення конфлікту між блоками валідним вважається той блок, який має більшу кількість "нащадків" (враховуючи форки).

Дві наступні роботи тих самих авторів і за цією ж тематикою мають назви SPECTRE та PHANTOM. У них також (не дуже вдало) просувається ідея побудови графчейну, причому запропоновані у роботах протоколи суттєво відрізняються між собою. Єдина спільна риса цих робіт – наявність грубих математичних помилок та відсутність строгих математичних обгрунтувань до своїх гучних заяв.

Крім описаних вище робіт, цікавими є також результати інших авторів, опубліковані у роботах Graphchain, Tangle, та деякі інші. Всі вони пропонують використовувати блокграф як узагальнення блокчейну. Основною метою таких робіт є збільшення потужності мережі за умови збереження її стійкості до атак. Ці протоколи, з одного боку, дійсно покращують характеристики протоколу Накамото, такі як висока пропускна спроможність, час підтвердження блоку, збереження лінійного порядку, тощо. Цікавим та багатообіцяючим видаються підходи, запропоновані у. Протокол, описаний у, пропонує використання "паралельних" ланцюжків блоків для обробки транзакцій. Такий підхід дійсно покращує пропускну здатність блокчейну, при збереженні стійкості до атак, проте час підтвердження блоку все одно залишається великим. Конструкція у роботі пропонує розділити блоки за їх базовими функціями – обробкою транзакцій, підтримка журналу, підтвердження блоку з транзакцією, тощо. Це дозволяє масштабувати кожен тип блоків окремо.

Розглянемо більш детально ці та інші роботи, що пропонують різні типи масштабування блокчейну. Найпершою роботою, в якій описується криптовалюта, що базується на графчейні (у ній використовується термін DAG), є блог-пост 2012 року. Ця робота описує криптовалюту DagCoin, побудовану без блоків, яка була презентована у 2015 році. Роль блоків тут відіграють окремі транзакції. Такий підхід не виявив якихось суттєвих переваг, проте започаткував нови тип протоколів консенсусу, які базуються на конструкції DAG.

Найпершою реально існуючою криптовалютою, яка використовує DAG замість блокчейну, є криптовалюта IOTA. Протокол консенсусу, що використовується для цієї валюти, називається Tangle (павутина). Згідно протоколу майнінга, кожен наступний блок посилається рівно на два попередніх. Для розв’язання конфліктної ситуації при наявності суперечливих транзакцій пропонується обчислити "висоту" конфліктуючих блоків та вибрати той, який знаходиться на більшій висоті. Такий підхід дає чудові можливості для атаки подвійної витрати, про що розробникам протоколу було повідомлено у приватному листуванні. Більш того, на конференції CryBlock-19 один з доповідачів по протоколу Tangle підтвердив, що на даний час не існує математично обґрунтованого доведення стійкості цього протоколу до атак, зокрема до атаки подвійної витрати. Тому виникає враження, що цей протокол не є стійким, просто зловмисники ще не виявили достатньої зацікавленості в атаках на нього.

2.3. Протоколи з контролем доступу

Hedera Hashgraph (Хешграф) є протоколом консенсусу, що базується на блокграфі. Цей протокол призначений для використання в середовищах з контролем доступу. Hashgraph є повністю асинхронним протоколом, що означає наявність властивостей стійкості та живучості без якихось додаткових допущень на час синхронізації. Hashgraph гарантує повне

вирішення проблеми Візантійської уголи за таких умов: не більше, ніж 1/3 ресурсів контролюється зловмисниками; зловмисник може видаляти повідомлення між чесними учасниками або затримувати їх на необмежений час. У той же час, для досягнення стійкості необхідно, щоб не менше за 2/3 від усіх учасників повинні весь час знаходитись в мережі; всі ці учасники мають бути чесними. Обробка транзакцій базується на неперервній синхронізації станів учасників при отриманні нової інформації щодо транзакцій або нової інформації про оновлення деякими учасниками інформації (“gossip about gossip” – "плітки про плітки"). Hashgraph має надзвичайно високу пропускну спроможність (близько 250 000 транзакцій в секунду [21]), але слід пам’ятати, що він діє лише у середовищі з контролем доступу. Така висока пропускна здатність є можливою завдяки відсутності значного перевантаження мережі: близько $n\log n$ для того, щоб транзакцію "побачили" всі учасники мережі.

Слабким місцем цього протоколу є те, що він орієнтований на середовища з контролем доступу. Це не дає можливості використовувати цей протокол (принаймні, існуючу версію цього протоколу) у різних застосуваннях, які вимагають повної децентралізації.

Нещодавно представлений протокол Casanova також орієнтований на використання у середовищах з контролем доступу. Подібно до більшості таких протоколів, Casanova є частково синхронним, що потребує гарантій його живучості. Однією з важливих характеристик цього протоколу є підтримка часткового порядку транзакцій замість повного, або лінійного, порядку. Тобто у цьому протоколі транзакції не є повністю упорядкованими. В протоколі Casanova блоки створюються влідаторами через регулярні часові інтервали. Консенсус досягається за кілька раундів. Він базується на голосуванні, яке виконують валідатори. Для оптимізації процесу досягнення консенсусу, Casanova підтримує кілька протоколів, використання яких залежить від ситуації. Тобто у різних умовах використовуються різні протоколи, як при умові наявності конфліктних транзакцій, так і за умови їх

відсутності. На сьогоднішній день, не існує інформації щодо практичного застосування протоколу Casanova.

2.4. Протоколи без контроля доступу

Протокол SPECTRE був представлений у 2016 році. В цьому протоколі новий блок розсилається відразу після створення. Кожен новий блок посилається на всі валідні блоки у графі, які він бачить на момент створення, і на які ще немає посилань. При наявності конфлікту протокол передбачає процедуру, яка нагадує "голосування". Неочікуваним є те, що у цій процедурі голосують не учасники, а самі блоки. Кожен блок у блокграфі голосує за один із блоків, у яких конфліктують транзакції. Легітимним вважається той блок (і, відповідно, та з конфліктуючих транзакцій), за який проголосувало більше блоків. Як саме блок повинен голосувати, визначається положенням цього блоку у графчейні. Такий протокол голосування у SPECTRE забезпечує швидке підтвердження блоків та добре масштабування, та лише слабку властивість живучості. Зловмисник має можливість підтримувати баланс між двома конфліктуючими транзакціями, тобто протокол є вразливим до узагальненої атаки розгалуження. Крім того, протокол голосування у SPECTRE прописаний нечітко, що допуская його неоднозначне тлумачення. Зокрема, це дає можливості і для атаки подвійної витрати, при виконанні певних додаткових умов, які є цілком імовірними. Робота, у якій презентується цей протокол, також не містить математично обґрунтованих тверджень стосовно оцінок імовірностей основних атак на блокчейн. Замість них навояться певні емпіричні міркування, які аж ніяк не можна вважати доведеннями.В третьому розділі ми розглянемо більш глибокий аналіз протоколу SPECTRE для демонстрації загальних властивостей блокчейнів на блокграфах.

На сьогоднішній день не відомо про якісь реальні застосування протоколу SPECTRE, що, зокрема, може свідчити про те, що автори і самі

розуміють його недостатню обґрунтованість. Невдовзі після SPECTRE, вони запропонували свій новий протокол.

У 2018 автори протоколу SPECTRE запропонували новий, зовсім інший протокол – так званий блокграф PHANTOM. Як стверджують автори, цей протокол розв’язує багато з зазначених вище проблем блокграфу. Він зовсім не схожий на протоколи GHOST та SPECTRE. Замість роботи з окремими блоками або їх парами, новий протокол працює з так званими кластерами – множинами блоків, в яких блоки є "сильно пов’язаними" між собою. Інші блоки, навпаки, мають мало "зв’язків" з блоками, які належать кластеру.Вважається, що блоки, які створюються чесними майнерами, будуть "сильно" пов’язаними з іншими блоками, які теж створені чесними майнерами. Тому припускається, що найбільший кластер, з переважаючою імовірністю, буде містити лише блоки, створені чесними майнерами. Також наводиться протокол побудови лінійного порядку, який суттєво залежить від вибраного основного кластера.

Протокол консенсусу полягає у пошуку найбільшого (або майже найбільшого) кластера. Протокол є досить складним та суперечливим. Оскільки він також не описаний чітко та допускає двозначне тлумачення, то можна змоделювати ситуацію, при якій виконання цього протоколу призведе до зациклення. Або при якій протокол не включить до кластеру блок, який має "сильні" зв’язки з іншими блоками кластера. Слід зазначити, що задача знаходження "правильного" кластеру є NP-повною, тому у реалізації протоколу пропонується використовувати лише деякий спрощений розв’язок задачі. Це й призводить до некоректності роботи протоколу. Автори PHANTOM проголошують високу пропускну здатність протоколу та наявність лінійного порядку, проте час підтвердження транзакції, а також час стабілізації блокграфу є дуже довгим. Вважається, що блок є стабільним тоді, коли на нього посилається деякий блок спеціального вигляду, що має назву "пісчаний годинник". Цей блок має бути таким, що посилається на всі блоки основного кластеру. Автори намагались оцінити імовірність появи такого

блоку та, відповідно, середнього часу до його очікування. Для цього вони застосовували апарат теорії імовірності, проте з величезними математичними помилками. Проте їм все одно не вдалось отримати якісь конкретні оцінки часу очікування такого блоку, крім того, що він є скінченним, оскільки імовірність появи такого блоку ненульова.

Для зменшення часу очікування консенсусу автори також пропонують застосувати деякий симбіоз цих двох протоколів – спочатку побудувати основний кластер з використанням протоколу PHANTOM, а потім застосувати протокол SPECTRE для встановлення лінійного порядку на блоках цього кластеру. Ця пропозиція ще не достатньо розроблена і зараз лише досліджується.

Протокол Graphchain був запропонований Boyen та іншими авторами, а згодом доповнений у. Протокол обіцяє не тільки вирішення питання масштабування, але й зменшення олігополії великих майнінгових пулів. На відміну від протоколів SPECTRE та PHANTOM, Graphchain працює відразу з окремими транзакціями, не об’єднуючи їх у блоки. Кожна транзакція посилається на вибрані "батьківські" транзакції, тобто ті, які є валідними і були створені та оброблені раніше. Для обробки кожної транзакції потрібно виконати деяку роботу, тобто цей протокол є PoW-подібним. Алгоритм вирішення конфліктної ситуації між транзакціями подібний до того, що реалізований у Bitcoin. Для кожної з конфліктних транзакцій обчислюється її "висота" – величина, що дорівнює сумарній роботі (PoW) цієї транзакції та всіх інших, на яких ця транзакція посилається (безпосередньо чи опосередковано), тобто транзакцій, що є її "предками". Транзакція, що має найбільшу "висоту", вважається валідною, а відповідна конфліктуюча транзакція "знищується". Автори протоколу стверджують, що такий спосіб підтвердження транзакцій спонукає майнерів "не конфліктувати, а співпрацювати". Крім того, відносно невеликий об’єм PoW дозволяє зменшити олігополію майнінгових пулів. На відміну від Bitcoin, обробка кожної окремої транзакції в Graphchain винагороджується: верифікація

транзакції вважається корисною працею, яка має бути оплачена.

Наступні три протоколи виглядають перспективними з точки зору використання на практиці, хоча й не розв'зують всіх проблем мережі. В деякому розумінні вони базуються на схожих ідеях.

Протокол Prism поділяє блоки на кілька класів, згідно до їх функцій: пропозитори, воутери (тобто ті, що голосують) та блоки, що обробляють транзакції. Блоки-пропозитори посилаються на блоки з транзакціями, тим самим підтверджуючи ці блоки. Блоки-воутери вирішують, які саме з блоків-пропозиторів будуть відібрані для формування основного ланцюжка. Цей основний ланцюжок і буде визначати стан журналу транзакцій – як ланцюжок блоків у Bitcoin. Блоки-воутери об'єднуються у кілька у кілька різних ланцюжків (в залежності від значення хеш у цьому блоці), які функціонують паралельно. Така організація блоків дозволяє масштабувати різні функції блокчейну (обробка транзакцій, підтвердження і тд) паралельно. Автори стверджують, що пропускна здатність цього протоколу та його латентність можуть масштабуватись аж до фізичних обмежень на мережу, зі збереженням стійкості до основних атак.

Протокол Fruit Chain теж розділяє блоки за функціями. Частина блоків, з великим значенням PoW, формують основний ланцюжок, як у Bitcoin. Крім цих блоків, існують також блоки з суттєво меншим значенням PoW, які використовуються тільки для обробки транзакцій. Ці блоки називаються "фрукти". Блоки з основного ланцюжка підтверджують блоки- фрукти.

Стійкість такого протоколу можна доводити за тими ж принципами, що й для Bitcoin – достатньо довести відповідні твердження для основного ланцюжка. Одна з найсуттєвіших переваг Fruit Chain – маленький час очікування до обробки транзакцій, за рахунок великої кількості "маленьких" блоків-фруктів. Проте цей протокол ніяк не знижує час очікування до підтвердження транзакції, тобто після її першого підтвердження блоком з основного ланцюжка все одно потрібно чекати, поки вийде певна кількість

блоків з "великим" PoW.

Протокол Parallel Chains, нещодавно запропонований у, має характеристики, аналогічні до характеристик протоколів Fruit Chain та Prism. Однією з основних його переваг є те, що він може застосовуватись і для PoW, і для PoS. Згідно до протоколу, блоки організовані у декілька (від кількох десятків до кількох сотень) практично незалежних ланцюжків. Перший ланцюжок є синхронізаційним. Кожен його блок посилається тільки на попередній блок цього ж ланцюжка. Блоки іншого ланцюжка посилаються на попередній блок цього самого ланцюжка і на останній блок першого ланцюжка, який вони бачать на момент створення. Лінійний порядок блоків у цьому протоколі забезчується саме завдяки наявності синхнізаційного ланцюжка. Причому завдяки деяким додатковим деталям протоколу транзакції у блоках різних ланцюжків не перетинаються, внаслідок чого швидкість обробки транзакцій зростає прямо пропорційно кількості ланцюжків. Такий протокол дійсно забезпечує суттєво вищу пропускну спроможність, а обґрунтування його стійкості до основних атак є таким самим, як і в традиційному блокчейні – для кожного ланцюжка окремо. Проте, знову-таки, цей протокол аж ніяк не дає можливості зменшити період очікування стабілізації графу, зокрема період очікування повного підтвердження транзакції.

У нещодавніх дослідженнях нам вдалось запропонувати новий протокол консенсусу на блокграфі, який у випадку конфліктних транзакцій є повним аналогом правила довшого ланцюжка. Для такого протоколу повністю обґрунтована стійкість до атаки подвійної витрати, проте стійкість до інших атак ще не досліджувалась. Такі дослідження можуть стати темою подальшої роботи.

Підсумки розділу

Отже, останніми роками активно досліджується можливість переходу з блокчейну на більш загальну структуру, яка має додаткові можливості, але

при цьому залишається стійкою до основних атак. В основному в якості нової структури розглядається блокграф (DAG), який має певні особливості. Роботи пропонують різні шляхи таких узагальнень та частковий аналіз їх властивостей. В той же час, до результатів таких досліджень виникає велика кількість питань. Виділимо основні з них.

1. Алгоритми вирішення конфлікту є досить сумнівними, з точки зору практичності та забезпечення стійкості до атак. В багатьох роботах вони навіть не описані коректно.

2. Неможливість (або лише часткова можливість) встановити лінійний порядок на множині блоків. Деякі протоколи встановлюють лише частковий порядок, деякі взагалі обходять це питання.

3. Довгий час очікування повного підтвердження транзакції, тобто часу від моменту включення транзакції у блок до того, як транзакція стане незворотною з великою імовірністю.

4. Недостатнє математичне обґрунтування основних тверджень, особливо стосовно стійкості протоколів до основних атак. Так, математичні перетворення у багатьох роботах містять суттєві помилки, а в інших замість математичних викладок наводяться евристичні міркування.

Тому поле для досліджень у цьому напрямку залишається надзвичайно великим, а задача масштабування блокчейну – дуже актуальною.

3. УДОСКОНАЛЕННЯ ПРОТОКОЛУ КОНСЕНСУСУ SPECTRE ТА ОБҐРУНТУВАННЯ СТІЙКОСТІ ОТРИМАНОЇ МОДИФІКАЦІЇ ДО АТАКИ ПОДВІЙНОЇ ВИТРАТИ

Нове покоління протоколів консенсусу, що пропонується останні роки багатьма спеціалістами у галузі блокчейну, базується на дещо новій концепції – використання блокграфу замість блокчейну. Всі запропоновані блок графи мають структуру орієнтованого ациклічного графу (DAG – directed acyclic graph). До таких протоколів належать GHOST (найперший з них), SPECTRE, PHANTOM, Graphchain, Tangle та багато інших. У таких протоколах блоки утворюють DAG як розподілену фінансову книгу. Основна причина таких наполегливих спроб створення альтернативної блок-структури полягає у надзвичайно низькій пропускній спроможності блокчейну. Наприклад, у мережі Біткойн, за різними підрахунками, обробляється від 3 до 7 транзакцій за секунду, що є просто мізерною величиною у порівнянні зі швидкістю банківських операцій. Враховуючи те, що один блок у БІТКОЙН виходить у середньому раз на 10 хвилин, а кількість блоків підтвердження вважається рівною 6 блокам, то, навіть якщо транзакція була оброблена миттєво, до її остаточного підтвердження потрібно чекати приблизно годину. Така ситуація робить неможливим використання криптовалюти для різних побутових операцій – ніхто не буде чекати годину, щоб оплатити біткоїнами чашку кави.

Тому основною метою розробки нових протоколів консенсусу є, перш за все, максимальне збільшення пропускної здатності мережі, або, що практично те ж саме, збільшення швидкості обробки транзакцій. При цьому основні характеристики мережі, що забезпечують її стійкість до атак різних типів, повинні бути збережені.

Найпростішими шляхами для вирішення такого питання видаються такі модифікації:

- збільшення обсягу блоку, тобто максимально можливої кількості транзакцій у ньому;

- зменшення інтервалу між виходом блоків.

Проте, на жаль, обидва таких підходи суттєво знижують стійкість мережі, у першу чергу до атаки подвійної витрати. Як було показано у, при суттєвій зміні цих параметрів зловмисник може провести атаку подвійної витрати з імовірністю 1 навіть тоді, коли його обчислювальні потужності суттєво менші за 50%.

Деякі з запропонованих протоколів консенсусу на графах дійсно покращують певні характеристики, у порівнянні з класичним "блокчейновим" протоколом консенсусу, запропонованим Накамото у. Так, використовуючи різні типи блокграфів, дійсно можна підвищити швидкість обробки транзакцій. Проте "платою" за таке покращення є виникнення у таких протоколах суттєвих недоліків, основні з яких є наступними:

- зниження стійкості до атак (в першу чергу до атаки подвійної витрати та до атаки розщеплення мережі);

- зменшення "стабільності" утвореного блокграфу;

- відсутність лінійного порядку на блоках (можливо задати лише частковий порядок), тощо.

Крім того, суттєво підвищується складність побудови обґрунтованих оцінок стійкості таких протоколів. На даний момент, в усіх роботах були знайдені суттєві математичні помилки у твердженнях, що стосуються оцінок імовірності атак.

У цьому розділі розглянемо протокол консенсусу SPECTRE. Це один з перших DAG-протоколів. Згідно протоколу, новий блок має містити посилання на всі "свіжі" блоки (тобто ті, на які ще немає посилань), які називають листками. Майнер створює новий блок, включаючи у нього хедери усіх листків, та відразу розсилає його всім іншим вузлам. Також цей протокол має дуже незвичний алгоритм вибору між конфліктуючими транзакціями у випадку зловмисного або випадкового форку. Згідно цього алгоритму, блоки графу ніби "голосують", в залежності від їх розташування у

графі, за одну з конфліктуючих транзакцій. Далі будемо детально аналізувати саме цей алгоритм.

Протокол SPECTRE забезпечує швидку обробку транзакцій та їх швидке підтвердження, а також може бути досить просто масштабований. Проте він гарантує лише слабку властивість життєздатності. Автори стверджують, що цей протокол є стійким до різних атак, наприклад до атаки подвійної витрати та атаки цензурування, але не дають ніяких доведень для цих тверджень, лише сумнівні емпіричні міркування.

У цьому розділі розглянемо дещо підсилену атаку подвійної витрати, яка на певному етапі використовує цензурування. Протягом цієї атаки зловмисник не лише будує альтернативні підграфи, але й вибирає, які саме блоки він буде підтверджувати. Така атака є більш ефективною за кожен з цих методів окремо і її імовірність може бути значною. Покажемо також, що імовірність успіху цієї атаки можна суттєво знизити, якщо трохи змінити правила прийому транзакції вендором.

У наступних пунктах цього розділу наведемо, строго обґрунтовані, формули для обчислення верхніх оцінок такої імовірності в залежності від параметрів мережі та кількості блоків підтвердження, які вендор має побачити до того, як вважатиме транзакцію незворотною. Для більшої наочності, будуть також наведені відповідні чисельні результати.

3.1. Опис протоколу консенсусу SPECTRE

SPECTRE є Proof-of-Work протоколом, у якому блоки впорядковані у вигляді DAG $G=(C, E)$, де C є множиною блоків (вершин графу), а E є множиною відповідних геш-посилань (ребер графу). Ребро-стрілочка ставиться від блока z_1 до блока z_2 тоді і тільки тоді, коли блок z_1 містить у своєму хедері геш-значення блоку z_2.

Правила майнінгу є дуже простими:

(1) Як тільки майнер створив новий блок або отримав його від іншого майнера, він повинен розповсюдити його у всій мережі.

(2) При створенні блоку, вкласти в його хедер список, що містить геш-значення від всіх "листків", які він "бачить".

Щоб запобігти виникненню конфліктуючих транзакцій, застосовується процедура, яку умовно можна назвати "голосуванням". При цьому кожен блок $z \in C$ має рівно один голос, який він віддає за одну із цих транзакцій. Щоб описати процес голосування, нам потрібно ввести деякі позначення з [12]:

- $past(z,G) \subset C$ означає множину усіх блоків (вершин графу), до яких існує шлях від блоку z;
- $future(z,G) \subset C$ означає множину усіх блоків (вершин графу), від яких існує шлях до блоку z;
- $cone(z,G) \subset C$ означає множину усіх блоків (вершин графу), які належать або до $past(z,G) \subset C$, або до $future(z,G) \subset C$;
- $virtual(G)$ означає деякий гіпотетичний блок цього графу, такий, що $past(virtual(G)) = G$.

Якщо в деяких двох блоках $x, y \in G$ наявні конфліктуючі транзакції, то потрібно застосувати процедуру голосування, щоб визначити, який з блоків (а, отже, і яка з транзакцій) має пріоритет. . Якщо деякий блок голосує за блок x, то це позначається як $x \prec y$, а відповідний голос блоку дорівнює -1 Якщо ж деякий блок голосує за блок y, то це позначається як $y \prec x$, а відповідний голос блоку дорівнює +1. Деякі блоку можуть приймати невизначене рішення, тоді голос дорівнює 0. Остаточне рішення приймається більшістю голосів (у випадку невизначеності обирається довільне рішення).

Правила голосування, згідно [12], є наступними:

(1) якщо для блоку $z \in G$ виконано: $z \in future(x)$ $z \notin future(y)$, то цей

блок голосує за $x \prec y$, тобто його голос дорівнює -1;

(2) якщо $z \in future(x) \cap future(y)$, то голос цього блоку визначається рекурсивно, а саме згідно з більшістю голосів у підграфі, утвореному його минулим (інакше кажучи, такий блок голосує як $virtual(past(z))$;

(3) якщо $z \notin future(x) \cup future(y)$, то цей блок голосує як більшість у $future(z)$;

(4) якщо $z = virtual(G)$, то він голосує так, як більшість блоків у його минулому (тобто я к більшість блоків у G);

(5) нарешті, якщо $z = x$ або $z = y$, то такий блок голосує сам за себе, наслідуючи кожен блок з свого минулого та створюючи прецедент для блоків, які не потрапили у його минуле.

У оригінальній роботі автори потоколу SPECTRE розглянули два типи атак: атака подвійної витрати та атака цензурування. При виконанні атаки подвійної витрати, зловмисник порушує перше правило майнінгу: після створення блоку він його деякий час не поширює, а потім поширює одночасово два підграфи, які містять по одній з конфліктуючих транзакцій. При виконанні атаки цензурування, зловмисник порушує друге правило: він підтверджує не всі листки, а тільки деякі з них вибірково, а саме ті блоки, які пов’язані з конфліктуючою транзакцією, яку він хоче нав’язати.

Розглянемо далі гібридну атаку, при виконанні якої зловмисник порушує зразу обидва правила майнінга.

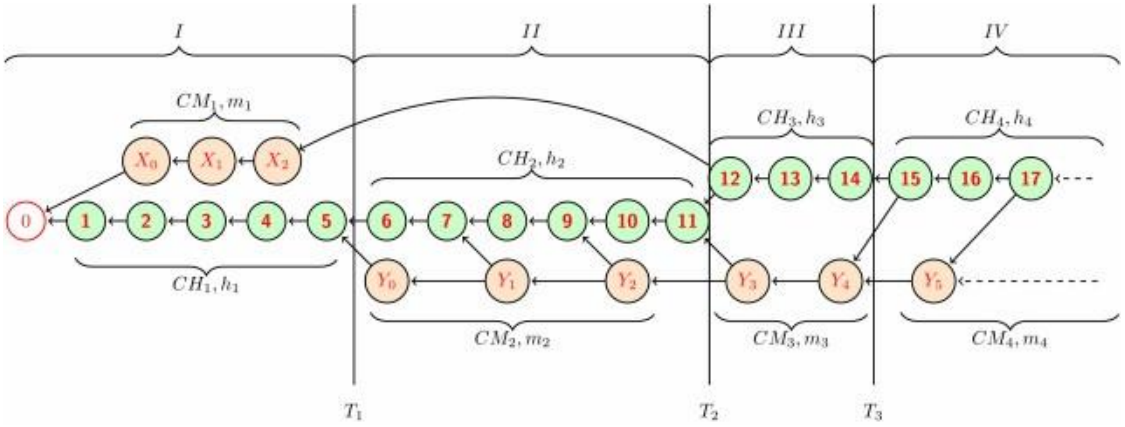
3.2. Опис гібридної атаки на протокол SPECTRE

Для зручності пояснення, розіб’ємо цю гібридну атаку на кілька етапів, що відрізняються певними діями зловмисника.

Етапи атаки розділені моментами часу T_1, T_2 та T_3. Фізичний зміст

цих моментів пояснюється нижче.

Для спрощення викладення, на цьому рисунку зображено лише деяку невелику кількість блоків, об’єднаних у декілька ланцюжків. Не зважаючи на це, при побудові атаки та при обчисленні її імовірності будемо вважати, що на кожному етапі атаки може бути створена довільна кількість блоків. Також слід зазначити, що якщо якийсь з ланцюжків замінимо на відповідне "дерево", то ні у самій схемі атаки, ні у обчисленнях та доведеннях нічого не зміниться.



Отже, атака складається з таких етапів.

Eman 1. Починається з того моменту, коли зловмисник створив одну з двох конфліктуючих транзакцій (якою пересилає монету до вендора та яку потім планує відмінити):

- зловмисник створює одну з двох конфліктуючих транзакцій та включає її до блоку X_0, після чого підтверджує цей блок іншими блоками, створюючи підграф (на рисунку – ланцюжок) CM_1 довжиною m_1; цей підграф він не розповсюджує, і взагалі нікому не пересилає;

- чесні майнери у цей час створюють підграф (на рисунку – ланцюжок) CH_1 довжиною h_1.

Eman 2. Починається з моменту T_1, коли зловмисник вирішує, що він створив достатню кількість блоків підтвердження для блоку X_0:

- зловмисник створює другу з конфліктуючих транзакцій (якою пересилає ту ж саму монету до свого гаманця) та включає її до блоку Y_0, після чого підтверджує цей блок іншими блоками, створюючи підграф (на рисунку – ланцюжок) CM_2 довжиною m_2; це підграф він також не розповсюджує, і взагалі нікому не пересилає;

- чесні майнери у цей час створюють підграф (на рисунку – ланцюжок) CH_2 довжиною h_2, який є продовженням підграфа CH_1.

Eman 3. Починається з моменту T_2, коли зловмисник вирішує, що він створив достатню кількість блоків підтвердження для блоку Y_0:

- зловмисник розповсюджує підграф CM_1;

- після того, як вендор побачив підграф CM_1 зі "своєю" транзакцією, він чекає деякий (можливо, нульовий) час Δ і потім надсилає товар зловмиснику;

- за час очікування чесні майнери створюють підграф (на рисунку – ланцюжок) CH_3 довжиною $h_3 \geq 0$ (у даній модифікації будемо вважати, що $h_3 \geq z$ для деякого прийнятного z, сенс якого буде пояснено нижче);

- зловмисник за цей час створює підграф (на малюнку – ланцюжок) CM_3 довжиною m_3, яким продовжує підтверджувати тільки блок Y_0; при цьому він поки що не розповсюджує підграфи CM_2 та CM_3.

Eman 4. Починається з моменту T_3, коли вендор відправляє товар і зловмисник про це дізнається:

- вендор відправляє товар у момент T_3;

- у той самий момент зловмисник розповсюджує підграфи CM_2 та CM_3;

- чесні майнери підтверджують обидва підграфи, CH_3 та CM_3, створюючи підграф CH_4 з h_4 блоків;

- зловмисник продовжує підтверджувати тільки підграф CM_3 підграфом CM_4 з m_4 блоків (комбінуючи атаку подвійної витрати з атакою

цензорування).

Атака буде успішною, якщо за блок Y_0 проголосує більше блоків, ніж за X_0.

Згідно схеми атаки та протоколу голосування, визначеному у SPECTRE, голоса блоків розподіляться наступним чином:

• усі блоки з CM_1 голосують за X_0 (m_1 блоків);

• усі блоки з CH_3 голосують за X_0 (h_3 блоки, а у даній модифікації – $h_3 \geq z$ блоків);

• усі блоки з CM_2, CM_3 та CM_4 голосують за Y_0 ($m_2 + m_3 + m_4$ блоків);

• усі блоки з CH_1 та CH_2 голосують як більшість у їх майбутньому ($h_1 + h_2$ блоки);

• усі блоки з CH_4 голосують як більшість у їх минулому (h_4 блоки).

Тепер пояснимо, у чому полягає вразливість вихідного протоколу з алгоритмом вирішення конфлікту. Згідно протоколу якому повинен поводитись вендор, щоб знизити ризики зникнення транзакції з оплатою за товар. Лише сказано, що він повинен дочекатись певної кількості блоків підтвердження, що посилаються на блок з відповідною транзакцією. При цьому не наводиться, як визначити достатню кількість блоків підтвердження та від чого вона залежить. Автори (без будь- якого доведення) стверджують, що імовірність атаки спадає експоненційно із зростанням кількості блоків підтвердження.

Тепер розглянемо етапи атаки. Не випадково зловмисник розповсюджує підграф CM_1 лише після того, як блок X_0 отримав достатню кількість блоків підтвердження. У цьому випадку вендор побачить практично одночасово всі блоки цього підграфу і, згідно протоколу, може одразу відправляти товар. Якщо зловмиснику пощастить і вендор відправить товар практично миттєво (у цьому випадку $h_3 = m_3 = 0$), для успіху атаки достатньо

виконання нерівності $m_2 > m_1$, а забезпечити її виконання повністю в межах можливостей зловмисника, незалежно від його обчислювальних потужностей.

Запропонована нами модифікація полягає у тому, що вендор повинен чекати створення певної кількості блоків підтвердження починаючи з того моменту, коли він вперше побачив блок з відпвідною транзакцією, навіть якщо у цей момент блок уже мав якусь, хай навіть дуже велику, кількість підтверджень. Така вимога гарантує виконання нерівності $h_3 \geq z$, де z – встановлена кількість блоків підтвердження (як її визначити буде пояснено далі). У цьому випадку, як буде показано у наступному пункті, імовірність атаки буде залежати від параметрів мережі (зокрема, від гешрейту зловмисника) та дійсно буде швидко (за експериментальними даними – експоненційно) спадати зі зростанням кількості блоків підтвердження.

3.3. Побудова верхньої оцінки імовірності гібридної атаки та визначення необхідної кількості блоків підтвердження

У цьому пункті доведемо теорему про верхню оцінку імовірності гібридної атаки. Для подальшого викладення нам будуть потрібні наступні позначення та допоміжні твердження, більшість з яких наведено у.

Нехай α_H, $\alpha_M > 0$ – величини, що характеризують інтенсивності генерації блоків чесними майнерами та зловмисником, відповідно, а $\alpha = \alpha_H + \alpha_M$ – загальна інтенсивність виходу блоків. Тоді ці величини будуть параметрами функцій розподілу випадкових величин T_H та T_M, що визначають, відповідно, час до виходу наступного блоку для чесних майнерів та для зловмисника:

$$F_{T_H} = P(T_H < t) = 1 - e^{-\alpha_H t}, \qquad F_{T_M} = P(T_M < t) = 1 - e^{-\alpha_M t}.$$

Тоді імовірності того, що наступний блок буде створено, відповідно, чесними майнерами та зловмисником, дорівнюють

$$p_H = \frac{\alpha_H}{\alpha} \qquad \text{та} \qquad p_M = \frac{\alpha_M}{\alpha}. \tag{3.2}$$

Позначимо D_H – найбільший час, який потрібно чесному майнеру, щоб розіслати вже створений блок всім (принаймні, всім чесним) майнерам (час синхронізації мережі).

Зробимо припущення на користь зловмисника і будемо вважати, що відповідний час $D_M = 0$, а також що зловмисник має можливість затримувати блоки, які розповсюджують чесні майнери, на будь-який час, що не перевищує D_H. Також він може затримувати свої блоки на будь-який час, а також може розповсюджувати їх миттєво, і взагалі може порушувати всі правила майнінгу.

Також визначимо імовірності наступних подій:

p_H' – імовірність того, що чесні майнери створили та розповсюдили наступний блок швидше, ніж це зробив зловмисник;

p_M' – імовірність альтернативної події.

Тоді, згідно Леми 1 у [24], отримуємо наступні рівності:

$$p_H' = e^{-\alpha_M D_H} p_H \qquad \text{та} \qquad p_M' = 1 - p_H' = 1 - e^{-\alpha_M D_H} p_H$$

Позначимо $P_z(k)$ імовірність того, що зловмисник створить рівно k блоків за той час, поки чесні майнери створять та розішлють z блоків. Тоді, згідно Леми 4 у [24],

$$P_z(k) = \sum_{i=0}^{k} \left\lceil C^i_{z+i-1} p_H^z p_M^k e^{-\alpha_M z D_н} \cdot \frac{(\alpha_M z D_H p_H)^{k-i}}{(k-i)!} \right\rceil.$$

Зазначимо, що у випадку $D_H = 0$ формула (3.4) виглядає значно простіше:

$$P_z(k) = C^k_{z+k-1} p_H^z p_M^k,$$

оскільки в цьому випадку у сумі (3.4) залишиться лише один доданок при $k = i$, а всі інші зникають через наявність множника $(\alpha_M z D_H p_H)^{k-i}$.

Тепер сформулюємо та доведемо основний результат.

Теорема 3.1. Нехай згідно до правил прийняття транзакції вендор чекає на отримання z блоків підтвердження, які з'явились після того, як він вперше побачив блок з цією транзакцією. Тоді, у минулих позначеннях, верхня границя імовірності $P_z(\alpha_M, \alpha_H, D_H)$ успіху гібридної атаки дорівнює:

$$P_z(\alpha_M, \alpha_H, D_H) \le 1 - \sum_{k=0}^{z-1} P_z(k) \cdot \left(1 - \left(\frac{p_M'}{p_H'}\right)^{z-k} \cdot \left(\left(p_H'\right)^{z-k-1} + 1\right)\right), \qquad (3.6)$$

зокрема у випадку, коли $D_H = 0$:

$$P_z(\alpha_M, \alpha_H, 0) \le 1 - \sum_{k=0}^{z-1} p_H^z p_M^k \cdot C_{z+k-1}^k \cdot \left(1 - \left(\frac{p_M}{p_H}\right)^{z-k} \cdot \left(\left(p_H\right)^{z-k-1} + 1\right)\right). \qquad (3.7)$$

При цьому, якщо $p_M' \ge p_H'$, то $P_z(\alpha_M, \alpha_H, D_H) = 1$ при будь-якому значенні $z \in \mathrm{N}$.

Доведення. Нехай B_1 є першим блоком, який містить обидва блоки X_0 та Y_0 у своєму минулому. Такий блок, а також усі блоки у його майбутньому будуть голосувати як більшість блоків у їх минулому. Отже, голоси блоків з підграфа CH_4 суттєво залежать від того, як проголосують блоки, які належать до $past(B_1, G)$. Проаналізуємо, як ці блоки можуть проголосувати:

усі блоки з CH_3 проголосують за X_0;

усі блоки з CM_3 проголосують за Y_0;

останній блок у CH_2, позначимо його B_2,

проголосує за Y_0, якщо $h_3 \le m_3$, і у цьому випадку так само проголосують усі блоки з CH_1 та CH_2.

Отже, однією з умов успішної атаки є умова $h_3 \le m_3$, яку далі будемо позначати умова ($C1$).

Якщо ж (C1) не виконується, зловмисник все одно може здійснити успішну атаку у випадку, якщо блоки у $past(B_2, G)$ проголосують за Y_0.

Проаналізуємо, коли таке можливо. Якщо (С1) не виконується, тоді B_2 голосує за X_0. Нехай $h_3 = m_3 + l$ для деякого $l \in \mathrm{N}$. Тоді блок, розміщений безпосередньо перед B_2, назвемо його B_3 (на Рис. 1 це блок 10), проголосує за Y_0 лише у тому випадку, якщо протягом часу між появою блоків B_3 та B_2 зловмисник створив не менше за $l+1$ блок. Аналогічно, якщо B_3 та B_2 голосують за X_0, необхідною умовою того, щоб того, щоб блок B_4, розміщений безпосередньо перед B_3, проголосував за Y_0, є умова, щоб протягом часу між появами блоків B_4 та B_3 зловмисник створив не менше за $l+2$ блоки, і т.д.

Отже, якщо умова (С1) не виконується, необхідною умовою для того, щоб деякий блок з підграфу CH_2 проголосував за Y_0, є така умова:

- у підграфі CH_2 існує такий блок (назвемо його B_5), що за час між появою цього блоку та наступного за ним зловмисник створив не менше за $l+1+u$ блоків, де $u = \#\left\{ CH_2 \cap future\left(B_5, G\right) \right\}$.

Цю умову позначимо (С2).

Якщо ж жодна з умов (С1) та (С2) не виконується, зловмисник все одно має шанс реалізувати свою атаку. Необхідною умовою для цього є така умова:

- у деякий момент часу $T > T_3$ буде виконуватись нерівність $m_4 = h_4 + l$.

Цю умову позначимо (С3).

Отже, для успішної реалізації атаки необхідно, щоб виконувалась принаймні она з умов (С1), (С2) або (С3).

Зазначимо, що, оскільки час до виходу блоків має експоненційний розподіл, то кількість блоків, створених за певний період, описується розподілом Пуасона. Тому процеси генерації блоків на різних часових інтервалах та на різних підграфах є незалежними. Звідси випливає незалежність подій, що полягають у виконанні умов (С1), (С2) та (С3).

За умовою теореми, $h_3 \geq z$. Тоді, використовуючи Лему 4 з [14],

отримаємо:

$$P(C1) = 1 - \sum_{k=0}^{z-1} P_z(k). \tag{3.8}$$

Далі,

$$P(\neg(C1) \cap (C2)) = P(\neg(C1)) \cdot P((C2)) =$$

$$= \sum_{k=0}^{z-1} P^z(k) \left(\left(p^{M'}\right)^{z-k} + \left(p^{M'}\right)^{z-k+1} + \ldots + \left(p^{M'}\right)^{z-k+(k-1)} \right) \leq \tag{3.9}$$

$$\leq \sum_{k=0}^{z-1} P_z(k) \left(\frac{\left(p'\right)_{z-k}}{1 - p_M'} \right) = \sum_{k=0}^{z-1} P_z(k) \left(\frac{\left(p_M'\right)_{z-k}}{p_H'} \right).$$

Крім того, справедливою є рівність

$$P(\neg(C1) \cap \neg(C2) \cap (C3)) = (1 - P((C1))) \cdot (1 - P((C2))) \cdot P((C3)) \leq$$

$$\leq (1 - P((C1))) \cdot P((C3)).$$

Тепер зауважимо, що виконання умови (С3) означає, що зловмисник зміг "обігнати" чесних майнерів за кількістю створених блоків після того, як відставав від них на $z - k$ блоків. Тому

$$P((C3)) \leq \left(\frac{p_M'}{p_H'} \right)^{z-k}.$$

Тоді

$$P(\neg(C1) \cap \neg(C2) \cap (C3)) \leq \sum_{k=0}^{z-1} P_z(k) \left(\frac{p_M'}{p_H'} \right)^{z-k}. \tag{3.10}$$

Отже, застосовуючи (3.8)-(3.10), отримаємо:

$$P_z(\alpha_M, \alpha_H, D_H) \leq 1 - \sum_{k=0}^{z-1} P_z(k) + \sum_{k=0}^{z-1} P_z(k) \frac{\left(p^{M'}\right)^{z-k}}{p_H'} + \sum_{k=0}^{z-1} P_z(k) \left(\frac{p_{M'}}{p_H'} \right)^{z-k} =$$

$$= 1 - \sum_{k=0}^{z-1} P_z(k) \left(1 - \frac{\left(p_M'\right)^{z-k}}{p_H'} - \left(\frac{p_M'}{p^{H'}} \right)^{z-k} \right) =$$

$$= 1 - \sum_{k=0}^{z-1} P_z(k) \left(1 - \left(\frac{p_M'}{p_H'} \right)^{z-k} \left(\left(p_H' \right)^{z-k-1} + 1 \right) \right),$$

що завершує доведення теореми.

Зазначимо, що формулу також зручно використовувати для обчислення мінімальної кількості блоків підтвердження, необхідних для того, щоб імовірність гібридної атаки була меншою за деяку наперед задану величину.

3.3. Чисельні результати для імовірності успіху гібридної атаки

Тепер наведемо таблиці з чисельними результатами, отриманими з використанням формули. У Таблицях наведено найменше значення кількості блоків підтвердження, для яких імовірність гібридної атаки є не більшою за 10^{-3}, при різних значеннях параметрів мережі. Ці величини обчислені для двох різних значень інтенсивності виходу блоків:

для $\alpha = \frac{1}{600} = 0.00167$, як для Біткойн та для $\alpha = \frac{1}{60} = 0.0167$,

тобто в 10 разів більшої інтенсивності.

Зауважимо, що ці Таблиці суттєво відрізняються від Таблиць 4 та 5 в [24], в яких також наводяться найменші значення для кількості блоків підтвердження. Відмінність полягає у тому, що в роботі було досліджено "класичний" блокчейн, у той час як у цьому розділі розглядається блокграф з набагато складнішим протоколом консенсусу.

Для обчислення кількості блоків підтвердження використовуються формули перебором по z обчислювали

$$\min\left\{ z \geq 1 : P_z(\alpha_M, \alpha_H, D_H) \leq 10^{-3} \right\}$$

Найменше значення кількості блоків підтвердження, для яких імовірність гібридної атаки є не більшою за 10^{-3}, при різних значеннях долі гешрейту зловмисника та при $\alpha = \frac{1}{600} = 0.00167$.

p_M	D_H			
	0 с	15 с	30 с	60 с
0.1	6	6	7	7
0.15	9	9	10	10
0.20	14	14	14	15
0.25	21	21	22	24
0.3	33	35	36	40
0.35	60	65	69	81
0.4	137	154	174	228

Найменше значення кількості блоків підтвердження, для яких імовірність гібридної атаки є не більшою за 10^{-3}, при різних значеннях долі гешрейту зловмисника та при $\alpha = \frac{1}{60} = 0.0167$.

p_M	D_H			
	0 с	15 с	30 с	60 с
0.1	6	7	8	11
0.15	9	11	13	20
0.20	14	17	23	43
0.25	21	29	44	172
0.3	33	55	114	-----
0.35	60	139	-----	-----
0.4	137	203	-----	-----

Незаповнені клітинки означають, що імовірність атаки дорівнює 1 при будь-якій кількості блоків підтвердження.

Ми обирали значення часу синхронізації D_H так, щоб гарантовано охопити той проміжок, який часто виникає у реальних мережах. Наприклад, для мережі Біткойн найчастіше час синхронізації змінюється в межах від 0 до 20 секунд.

Також слід звернути увагу, що за умови $D_H = 0$ перший стовпчик у Таблицях однаковий, у той час як всі інші суттєво відрізняються. Пояснення цього факту полягає у наступному: чим більший час синхронізації чесних майнерів, тим більшу перевагу має зловмисник. Більш того, за умови ненульового часу синхронізації, перевага зловмисника суттєво залежить від інтенсивності генерації блоків: чим більша інтенсивність, тим більша перевага. Цей факт, зокрема, був аналітично доведений у для блокчейну,

тут також можемо побачити це у формулі: чим більшою є величина $\alpha_M D_H$, тим більшою є імовірність p_M'; а чим більша ця імовірність, тим меншою є так звана межа безпеки, тобто критична доля зловмисника (при $D_H = 0$ вона дорівнює 50%). Наприклад, як видно в останньому стовпчику Таблиці 3.2, при $D_H = 60$ секунд та $\alpha = \frac{1}{60} = 0.0167$, межа безпеки є меншою за 30%.

Підсумки розділу

За останні 5-7 років було запропоновано багато протоколів консенсусу на блокграфах. Слід зазначити, що використання таких протоколів, хоч і призводить до значного збільшення швидкості обробки транзакцій, проте несе у собі певні ризики. На даний час для жодного з таких протоколів немає строгих доведень стійкості до основних атак, таких як атака подвійної витрати, атака розщеплення мережі, тощо. Побудова таких доведень вимагає подолання великих аналітичних складнощів, навіть якщо використовувати дуже спрощену математичну модель мережі.

У цьому розділі розглянуто підхід до побудови оцінок стійкості протоколу консенсусу на блок графах, побудувано верхні оцінки для імовірності атаки подвійної витрати на протокол консенсусу SPECTRE. Для збільшення його захищеності, розглянуто дещо нестандартний алгоритм прийняття транзакції вендором: він повинен чекати, щоб певна кількість блоків підтвердження була створена, так би мовити, під його наглядом.

Важливими властивостями розглянутих результатів є те, що, крім самої оцінки імовірності, можна також обчислити необхідну кількість блоків підтвердження, достатню для того, щоб гарантувати необоротність транзакції з імовірністю, як завгодно близькою до 1.

Також слід зазначити, що метод побудови таких оцінок, запропонований тут, може бути застосований лише для протоколу SPECTRE, і не може бути легко перенесений на інші протоколи на блокграфах. Але сама ідея методу, можливо, може бути корисною і для інших протоколів.

4. СТІЙКІСТЬ ГЕШ-ФУНКЦІЙ, ЯКІ ВИКОРИСТОВУЮТЬСЯ У АНОНІМНИХ БЛОКЧЕЙНАХ, ДО СТАТИСТИЧНИХ АТАК

Блокові алгоритми шифрування, які, замість бінарних, використовують операції у простих скінченних полях, виникли у зв'язку з появою нових методів доведення без розголошення, які використовуються у деяких блокчейн-системах. Тут маємо на увазі так звані SNARK- та STARK-доведення. Побудова таких доведень починається з того, що певне перетворення (наприклад, геш-функцію) потрібно описати як систему певних рівнянь від багатьох змінних над скінченним полем, у лівій частині яких міститься поліном від багатьох змінних другого степеню, а у правій частині – поліном від багатьох змінних першого степеню. Ці рівняння називаються констрейнтами, і від їх кількості залежить складність побудови відповідного SNARK-доведення. Найчастіше SNARK-доведення використовуються для доведення знання прообразу деякої геш-функції. Тому геш-функції, що використовуються у таких блокчейнах, повинні бути спроектовані так, що їх можна було описати якомога меншою кількістю констрейнтів.

Однією з перших геш-функцій, зручних для побудови SNARK-доведень, була геш-функція Педерсена (Pedersen hash-function). Вона базується на операціях у групі точок еліптичної кривої, які, в свою чергу, можна звести до операцій у відповідному скінченному полі. Оскільки констрейнти є поліномами саме над таким полем, то кількість констрейнтів, необхідних для задання такої геш-функції, у десятки разів менше, ніж для "класичних" геш-функцій, що оперують з байтовими та бітовими операціями (близько 1.68 констрейнтів на 1 біт входу). Така кількість констрейнтів є досить прийнятною, проте питання про її зменшення все одно залишилось актуальним.

Нова геш-функція Poseidon дозволяє зменшити кількість констрейнтів до 15 разів, у порівнянні з геш-функцією Педерсена. У її побудові використовуються дві конструкції – так звана конструкція SPONGE (SPONGE construction) та новий блоковий

алгоритм шифрування HadesMiMC, який теж адаптований саме для використання у SNARK-доведеннях і тому базується на операціях у простих скінченних полях.

Для побудови геш-функції Poseidon алгоритм HadesMiMC використовується у конструкції SPONGE у якості внутрішньої випадкової перестановки. Тому криптографічна стійкість геш-функції Poseidon значною мірою визначається стійкістю цього алгоритму до основних криптографічних атак, які можна застосувати до таких алгоритмів.

Автори алгоритму HadesMiMC, а потім і автори геш-функції Poseidon виконали достатньо детальний аналіз стійкості зазначеного алгоритму до певного класу атак, який вони назвали “алгебраїчними атаками”. Проте заявлені ними оцінки стійкості до таких статистичних атак, як лінійні та різницеві, є здебільшого емпіричними, і, більш того, суттєво спираються на хибні твердження. Зокрема:

- не вказано, від яких саме параметрів залежить стійкість алгоритму до лінійних атак;

- помилково вважається, що координатні функції S-блоків визначають стійкість небінарного алгоритму до статистичних атак;

- неправильно обчислено кількість активних S-блоків алгоритму (без урахування його особливості, що полягає у наявності великої кількості раундів з лише одним S-блоком), тощо.

Крім того, автори стверджують, що оцінки стійкості до лінійного криптоаналізу отримуються аналогічно до різницевого. Проте, як було доведено у [38, 39], для небінарних алгоритмів, тобто таких, у яких ключовий суматор, блок підстановки та лінійний оператор використовують операції у скінченному простому полі, методи оцінювання стійкості до статистичних атак, особливо до лінійних, суттєво відрізняються від класичних, а також методи оцінювання стійкості до лінійного криптоаналізу дуже відрізняються від методів оцінювання стійкості до різницевого.

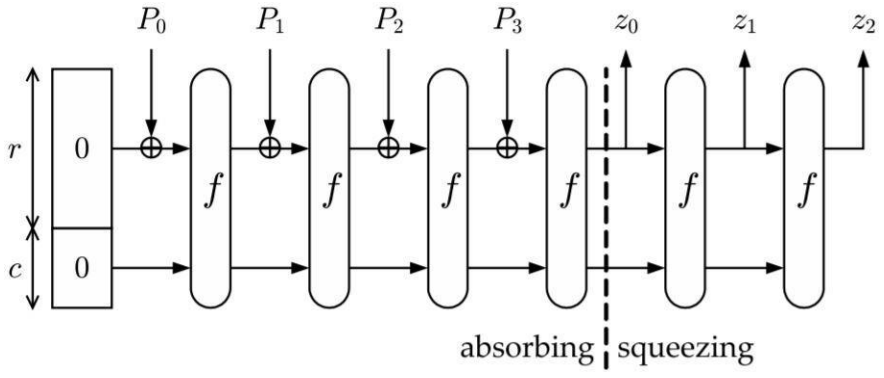
Виходячи з зазначеного, можна стверджувати, що питання стійкості до

статистичних атак як алгоритму HadesMiMC, так і геш-функції Poseidon залишається вікритим.

У цьому розділі наведемо, строго обґрунтовані, оцінки стійкості алгоритму блокового шифрування HadesMiMC (та, як наслідок, і геш-функції Poseidon) до лінійних та різницевих атак. Як проміжний результат, будуть отримані параметри, що залежать від S-блоків та визначають стійкість алгоритму до лінійних атак. Також для певних параметрів цього алгоритму, при яких він є сумісним з триплетами MNT-4 та MNT-6, що використовуються для рекурсивних SNARK-доведень, будуть наведені конкретні значення таких оцінок та показано, як визначити кількість раундів з повним шаром S-блоків, що гарантують достатній рівень стійкості цього алгоритму.

5.1. Математичні моделі алгоритму блокового шифрування HadesMiMC та геш-функції Poseidon

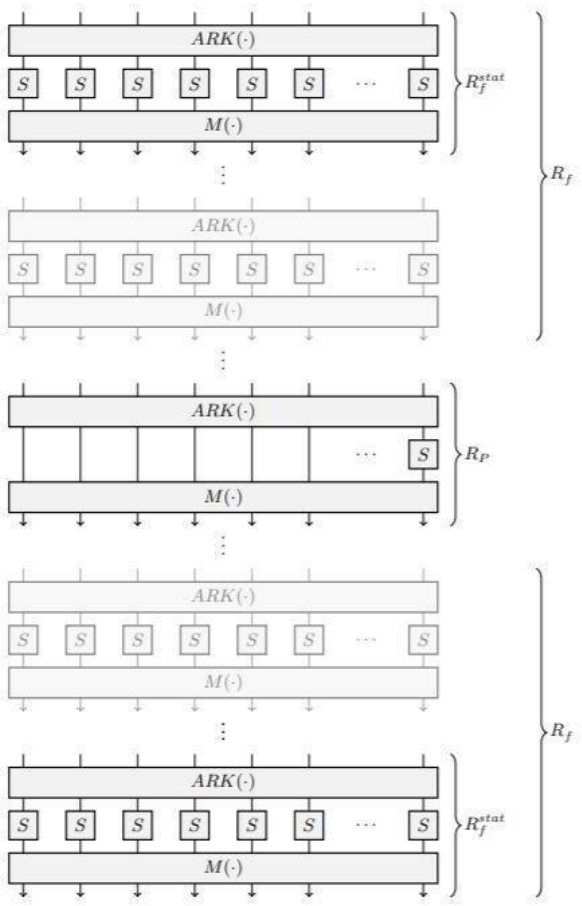
Геш-функція Poseidon використовує конструкцію SPONGE з внутрішньою перестановкою, що задається блоковим алгоритмом HadesMiMC.



Така конструкція задається трьома параметрами: ємністю c, швидкістю r та довжиною перестановки N, де $N = c + r$. Як частковий випадок, з

певних практичних міркувань, нас будуть цікавити параметри $N = 3\lceil \log p \rceil$, $c = 2\lceil \log p \rceil$, $r = \lceil \log p \rceil$. Як уже було зазначено, автори запропонували саме HadesMiMC у якості внутрішньої перестановки з метою зниження кількості констрейнтів на біт входу.

Для побудови оцінок стійкості до статистичних атак, розглянемо HadesMiMC як блоковий алгоритм шифрування, специфічною рисою якого є те, що його раундові функції відрізняються між собою. Основною ідеєю побудови цього алгоритму є використання раундових функцій двох типів: функцій з повним шаром S-блоків та функцій з частковим шаром S-блоків (наприклад, з лише одним S-блоком). Як буде показано далі використання такого алгоритму дозволяє суттєво зменшити кількість констрейнтів, зберігаючи при цьому прийнятний рівень стійкості до статистичних (а також алгебраїчних) атак.



Перейдемо до побудови математичної моделі HadesMiMC, яка є необхідною для аналізу його стійкості до зазначених атак.

Нехай p велике просте число, l його бітова довжина, $l \approx \log p$.

Визначимо бієктивні відображення $s : F_p \to F_p$ як $s(x) = x^u \bmod p$, де $(u, p-1) = 1$.

Для деякого $t \in N$ визначимо величини $x, C \in (F_p)^t$ як $x = (x_t, ..., x_1)$, $C = (c_t, ..., c_1)$, де $x_i, c_i \in F_p$, $i = \overline{1,t}$. Для $x \in (F_p)^t$ також визначимо два відображення, $S^{full} : (F_p)^t \to (F_p)^t$ та $S^{part} : (F_p)^t \to (F_p)^t$, як

$$S^{full}(x) = (s(x_t), ..., s(x_1)), \qquad S^{part}(x) = (x_t, ...x_2, s(x_1)).$$

Finally, define MDS-matrix $A : (F_p)^t \to (F_p)^t$ of the size $t \times t$.

Тепер визначимо раундові функції для алгоритму HadesMiMC. Як було зазначено, раундові функції у цьому алгоритмі є двох типів: раундові функції з повним шаром S-блоків, які визначаються як $f_C^{full} : (F_p)^t \to (F_p)^t$, де для довільного $C \in (F_p)^t$:

$$f_C^{full}(x) = A \circ S^{full}(x * C),$$

та раундові функції з частковим шаром S-блоків, які визначаються як $f_C^{part} : (F_p)^t \to (F_p)^t$, де для довільного $C \in (F_p)^t$:

$$f_C^{part}(x) = A \circ S^{part}(x * C),$$

де $x * C = (x_t + c_t, ..., x_1 + c_1)$, а "+" є операцією додавання у скінченному полі F_p (тобто операцією додавання за модулем p).

Означення 5.1. HadesMiMC-подібною перестановкою з параметрами p, t, u, r_{full} та r_{part} будемо називати сімейство перестановок $H_C^{(p,t,u,r_{full},r_{part})} : (F_p)^t \to (F_p)^t$, параметризованих множиною раундових

констант $C = \left(C_1,...,C_{2r_{full}+r_{part}}\right)$, $C_i \in \left(F_p\right)^t$, і визначених як

$$H_C^{(p,t,u,r_{full},r_{part})}(x) = f_{C_{2r_{full}+r_{part}}}^{full} \circ ... \circ f_{C_{r_{full}+r_{part}+1}}^{full} \circ f_{C_{2r_{full}+r_{part}}}^{part} \circ ... \circ f_{C_{r_{full}+1}}^{part} \circ f_{C_{r_{full}}}^{full} \circ f_{C_1}^{full}(x).$$

(

Якщо параметри p, t, u, r_{full} та r_{part} зафіксовані, ми, для спрощення запису, будемо використовувати позначення H_C.

Зауваження. Перетворення визначається так. Для фіксованих констант $C = \left(C_1,...,C_{2r_{full}+r_{part}}\right)$ спочатку застосовуємо до аргументу x функції $f_{C_1}^{full},..., f_{C_{r_{full}}}^{full}$, визначені у (з повним шаром S-блоків), з константами $C_1,...,C_{r_{full}}$ у якості відповідних "раундових ключів". Так виконуються перші r_{full} раундів цього перетворення. Потім застосовуємо функції, визначені у (5.3) (тобто функції з частковим шаром S-блоків), протягом r_{part} раундів, на відповідних “раундових ключах”. І, зрештою, знову застосовуємо r_{full} раундів з функціями.

5.2. Оцінки криптографічної стійкості алгоритму блокового шифрування HadesMiMC та геш-функції Poseidon

Як було зазначено, криптографічна стійкість конструкції SPONGE значною мірою визначається стійкістю її внутрішньої перестановки, у даному випадку – перестановки HadesMiMC. Так, у роботі було доведено, що якщо внутрішня перестановка конструкції SPONGE є нерозрізнювальною від випадкової перестановки, то відповідна геш-функція є нерозрізнювальною від випадкового оракула за умови, що кількість запитів до оракула не більша за $2^{\frac{c}{2}}$.

Для практичних застосувань зручно вибрати ємність як $c = 2l(p)$. При такому виборі параметру тобто рівень геш-функції дорівнює $l(p)$ (звичайно,

за умови "стійкої" внутрішньої перестановки). Отже, нам потрібно довести, що рівень криптографічної стійкості внутрішньої перестановки теж не менший за $l(p)$. Далі будемо використовувати цю вимогу для визначення кількості раундів з повним шаром S-блоків у перестановці HadesMiMC.

Далі побудуємо, строго обґрунтовані, оцінки криптографічної стійкості перестановки HadesMiMC до двох типів статистичних атак – лінійних та різницевих. Зазначимо, що при побудові оцінок стійкості до різницевого криптоаналізу будемо здебільшого використовувати відомі результати або їх узагальнення. Але для побудови оцінок стійкості до лінійних атак виявилось необхідним довести ряд нетривіальних тверджень про властивості характерів адитивної групи скінченного поля та про суми таких характерів.

5.2.1. Оцінки стійкості небінарного блокового алгоритму шифрування HadesMiMC до різницевого криптоаналізу

У подальшому викладенні будемо використовувати наступні допоміжні результати.

Означення 5.2 **:**

Блоковий шифр E з раундовою функцією

$$f : M \times K \to M$$

(тут M є абелевою групою з деякою операцією "$*$" та нейтральним елементом 0) називається *марковським шифром* по відношенню до операції "$*$", якщо $\forall x, \alpha, \beta \in M$:

$$\frac{1}{|K|}\sum_{k\in K}\delta\left(f(k, x*\alpha) * f(k,x)^{-1}, \beta\right) = \frac{1}{|K|}\sum_{k\in K}\delta\left(f(k,\alpha) * f(k,0)^{-1}, \beta\right),$$

де δ є символом Кронекера: $\delta(x,y) = \begin{cases} 1, & якщо\ x = y, \\ 0, & інакше. \end{cases}$

Зауваження: це означення можна легко узагальнити на випадок шифру з різними раундовими функціями.

Означення 5.3 **:**

Індекс розгалуження матриці $A:\left(F_p\right)^t \to \left(F_p\right)^t$ розмірності $t \times t$ визначається як

$$br\left(A\right) = \min_{x \in \left(F_p\right)^t \setminus (0,\ldots,0)} \left\{ wt\left(Ax\right) + wt\left(x\right) \right\},$$

де wt є вагою Хемінга.

Зазначимо, що якщо A є MDS-матрицею, то її індекс розгалуження є максимально можливим (для її розміру) і дорівнює $br\left(A\right) = t+1$. Наприклад, якщо $t = 3$, то $br\left(A\right) = 4$.

Твердження 5.1: блоковий шифр є марковським шифром.

Це твердження може бути доведено безпосередньо, перевіркою властивості для раундових функцій цього шифру.

Твердження 5.2 для марковського шифру (5.4), імовірність його різницевої характеристики обмежена зверху величиною Δ^b, де

$$\Delta = \max_{\alpha,\beta \in F_p^*} \frac{1}{p} \sum_{x \in F_p} \delta\left(s\left(x+\alpha\right) - s\left(x\right), \beta\right),$$

"+" є операцією додавання у полі, а b є кількістю активних S-блоків у всіх раундах разом.

Твердження 5.3: кількість активних S-блоків у двох послідовних раундах з раундовими функціями є не меншою за величину $br\left(A\right)$.

Якщо, зокрема, матриця A є MDS-матрицею розмірності $t \times t$, тоді $br\left(A\right) = t+1$ та, згідно Твердженню 5.3, кількість активних S-блоків у двох послідовних раундах з повним шаром S-блоків є не меншим за $t+1$. Але якщо між двома раундами з повним шаром S-блоків розміщуються кілька раундів з неповним шаром S-блоків, то неможна нічого стверджувати про кількість активних S-блоків у цих раундах, хіба що те, що їх не менше, ніж кількість раундів з повним шаром S-блоків. Слід зазначити, що автори якраз не взяли цього факту до уваги, і в результаті неправильно визначили

кількість активних S-блоків у алгоритмі, що призвело до отримання некоректних оцінок стійкості.

Твердження 5.4 : кількість активних S-блоків у всіх раундах алгоритму є не меншою за кількість активних S-блоків у раундах з повним шаром S-блоків.

Твердження 5.5 кількість b активних S-блоків у (5.4) є не меншою за

$$b \geq 2(t+1) \cdot \left\lfloor \frac{\lceil r_{full} \rceil}{2} \right\rfloor,$$

та, якщо r_{full} є парним, є не меншою від

$$b \geq (t+1) r_{full}.$$

Зауваження: з випливає, що використання парного значення величини r_{full} є більш ефективним, оскільки в даному випадку приєднання ще одного раунду, яке зробить цю величину непарною, не збільшить величину (5.7). Тому у подальшому викладенні вважаємо, що r_{full} приймає парні значення.

У цьому розділі розглянемо два випадки вибору S-блоків для HadesMiMC-подібного алгоритму: інверсні S-блоки та S-блоки, задані степеневими функціями. Перед отриманням основних результатів стосовно стійкості такого алгоритму, сформулюємо наступне допоміжне твердження.

Твердження 5.6.

1. Нехай $s(x) = x^u \bmod p$, де $(u, p-1) = 1$. Тоді $\Delta \leq \frac{(u-1)}{p}$.

2. Нехай $s(x) = \begin{cases} x^{-1} \bmod p, \; if \;\; x \neq 0; \\ 0, \; else. \end{cases}$. Тоді $\Delta \leq \frac{4}{p}$.

Теорема 5.1.

1. Нехай r_{full} є парним, $s(x) = x^u \bmod p$, $A : (F_p)^t \to (F_p)^t$ є MDS-матрицею розмірності $t \times t$. Тоді середня імовірність різницевої характеристики блокового шифру не перевищує величину

$$\left(\frac{u-1}{p}\right)^{(t+1)r_{full}}.$$

2. Нехай r_{full} є парним, $s(x) = x^{-1} \bmod p$, $A: (F_p)^t \to (F_p)^t$ є MDS-матрицею розмірності $t \times t$. Тоді середня імовірність різницевої характеристики блокового шифру (5.4) не перевищує величину

$$\left(\frac{4}{p}\right)^{(t+1)r_{full}}.$$

Доведення цієї теореми випливає відразу з Тверджень 5.1-5.6 та того факту, що в цьому випадку $\left\lceil \frac{r_{full}}{2} \right\rceil + \left\lceil \frac{r_{full}}{2} \right\rceil = r_{full}$. □

Зазвичай блоковий шифр вважається практично стійким до різницевого криптоаналізу, якщо імовірність його різницевої характеристики не перевищує 2^{-N}, де N є розміром блоку. Але у даному випадку максимальний рівень стійкості конструкції

SPONGE дорівнює $l(p) \approx \log p$. Тому можна сформулювати більш слабку вимогу як

$$\Delta^b < 2^{-\log p}.$$

Але, щоб збільшити стійкість та зробити її ближчою до теоретичної, будемо вимагати виконання нерівності

$$\Delta^b < 2^{-2\log p},$$

що означає

$$\left(\frac{u-1}{p}\right)^{(t+1)r_{full}} < 2^{-2\log p} \quad \text{або} \quad \left(\frac{4}{p}\right)^{(t+1)r_{full}} < 2^{-2\log p},$$

для степеневих та інверсних S-блоків, відповідно.

Крім того, додавши два додаткових раунди "про всяк випадок". Але додавання двох лопаткових раундів (один на початку, один у кінці) зробить r_{full} непарним і, як було показано раніше, не підвищить стійкості алгоритму. Тому якщо додавати додаткові раунди, то по

два раунди на початку і в кінці, щоб зберегти парність r_{full}.

Як можна бачити, перестановка зі степеневими S-блоками при $u > 5$ вимагає більше раундів, ніж перестановкм з інверсними, для того ж самого рівня стійкості. Далі обговоримо, який тип S-блоків має переваги з різних точок зору.

5.2.2. Оцінки стійкості небінарного блокового алгоритму шифрування HadesMiMC до лінійного криптоаналізу

Параметри, які характеризують практичну стійкість блокового шифру до лінійного криптоаналізу, суттєво залежать від структури цього шифру, зокрема від операції у ключовому суматорі. Так, оцінка середньої (за ключами) імовірності лінійної характеристики (по відношенню до операції додавання у полі F_p) шифру E не перевищує величину

$$\max_{\chi,\rho \in \hat{F_p}} ELP^E(\chi,\rho) = L^b,$$

де величина b дорівнює кількості активних S-блоків, а параметр L залежить від S-блоку:

$$L = L(s) = \max \frac{1}{} \sum \left(\chi(x), \rho(s(x)) \right) \Bigg|^2,$$

$$\left| \sum_{\chi,\rho \in \hat{F_p}} p \right| - \sum_{x \in F_p}$$

де χ та ρ є адитивними характерами поля F_p (тобто характерами адитивної групи цього поля).

Величина b повністю визначається лінійним (по відношенню до операції додавання у полі) оператором A і обчислюється так же само, як і при побудові імовірності різницевої характеристики, тобто b дорівнює найменшій можливій кількості активних S-блоків шифру. Застосовуючи такі ж міркування отримуємо $b = (t+1)r_{full}$, якщо r_{full} є парним числом (а тут розглядається саме такий випадок).

Теорема 5.2.

1. Нехай r_{full} парне, $s(x) = x^u \bmod p$, $A: (F_p)^t \to (F_p)^t$ є MDS-матрицею розмірності $t \times t$. Тоді середня імовірність лінійної характеристики шифру (5.4) не перевищує величини

$$\left(\frac{(u-1)^2}{p} \right)^{(t+1)r_{full}}.$$

2. Нехай r_{full} парне, $s(x) = x^{-1} \bmod p$, $A: (F_p)^t \to (F_p)^t$ є MDS-матрицею розмірності $t \times t$. Тоді середня імовірність лінійної характеристики шифру (5.4) не перевищує величину

$$\left(\frac{16}{p} \right)^{(t+1)r_{full}}.$$

Доведення: випливає із Тверджень 5.1-5.5, Твердження 5.7 та того факту, що в цьому випадку $\left\lceil \frac{r_{full}}{2} \right\rceil + \left\lceil \frac{r_{full}}{2} \right\rceil = r_{full}$. □

5.3. Вибір S-блоків

При виборі S-блоків потрібно враховувати наступні моменти:

- відображення $s: F_p \to F_p$ повинно бути бієктивним, тобто для

степеневих S-блоків повинна виконуватись вимога $\left(p-1,u\right)=1$;

- з міркувань стійкості до лінійного та різницевого криптоаналізу, параметри Δ та L (які повністю визначаються S-блоками) повинні бути якомога меншими;

- у термінах складності реалізації SNARK-доведень, кількість констрейнтів (яка залежить лише від кількості та типу S-блоків) повинна бути якомога меншою.

Нагадаємо, що для інверсних S-блоків параметри Δ та L обмежені

зверху величинами $\Delta \le \frac{4}{p}$ та $L \le \frac{16}{p}$, а для степеневих – величинами $\Delta \le \frac{(u-1)}{p}$ та $L \le \frac{(u-1)^2}{p}$ (Твердження 5.6 та 5.7). Тому для степеневих S-блоків має сенс вибирати параметр u як $u = \min\{v \in N : (v, p-1) = 1\}$.

Для інверсних S-boxes параметри Δ та L будуть гарантовано меншими, ніж для степеневих, якщо $u \neq 3$. Тому, з точки зору криптографічної стійкості, найбільш привабливими є інверсні та кубічні S-блоки (якщо тільки кубічні задають бієктивне відображення). Якщо $p-1$ ділиться на 3, тоді інверсні S-блоки не мають конкурентів (з точки зору криптографічних вимог).

З точки зору мінімізації кількості констрейнтів, інверсні та степеневі S-блоки з маленьким значенням параметра u також є найбільш привабливими. Дійсно, щоб задати інверсний S-блок, потрібно 3 контрейнти; щоб задати кубічний S-блок – 2 констрейнти:

$$\begin{cases} x_1 x_1 = x_2; \\ x_1 x_2 = x_3, \end{cases}$$

щоб задати S-блок $s(x) = x^5 \bmod p,$ потрібно 3 констрейнти:

$$\begin{cases} x_1 x_1 = x_2; \\ x_2 x_2 = x_3; \\ x_1 x_3 = x_4, \end{cases}$$

і т.д., в загальному випадку із зростанням показника u кількість констрейнтів зростатиме приблизно як $2\log u$. Отже, якщо $p-1$ ділиться на всі відносно малі прості 3, 5, 7, 11,…, тоді і з точки зору стійкості, і з точки зору складності реалізації SNARK-доведень, найкращими є інверсні S-блоки.

Проте, з іншого боку, реалізація перетворення інверсії є досить трудомісткою, оскільки вимагає застосування алгоритму Евкліда, який, в свою чергу, потребує порядку $O(\log p)$ ділень із залишком. Таким чином, при виборі S-блоків всі ці фактори повинні бути взятими до уваги, та певне

компромісне рішення має бути прийнято.

Далі, у розділі 5.5, при виборі параметрів алгоритму шифрування для заданого значення характеристики поля, буде розглянуто два варіанти вибору S-блоків – інверсні S-блоки та степеневі S-блоки, з найменшим значенням показника степеню, при яких S-блок задає бієктивні відображення.

5.4. Кількість раундів з повним та частковим шаром S-блоків та кількість констрейнтів, яка потрібна для задання шифру

Визначимо кількість раундів з повним шаром S-блоків, r_{full} , як мінімальну кількість раундів, що гарантує стійкість до різницевого та лінійного криптоаналізу (враховуючи, що крипто аналіз можна проводити і у одному напрямку, і у зворотньому). Після цього визначається кількість раундів з частковим шаром S-блоків, виходячи з міркувань стійкості до алгебраїчних атак.

Як було зазначено, автори також рекомендували додати два раунди про всяк випадок. Проте після додавання двох раундів (одного на початку, одного в кінці) величина r_{full} стане непарною, тобто додавання двох раундів не призведе до підвищення стійкості до статистичних атак. Якщо ж додавати по два раунди на початку, і у кінці, то це суттєво збільшить кількість констрейнтів. Не бачимо обґрунтованої необхідності збільшувати кількість раундів з повним шаром шаром S-блоків, особливо у ситуації, коли кількість констрейнтів є критичною величиною.

Кількість констрейнтів на біт визначається наступним чином. Кількість констрейнтів, необхідних для задання одного S-блоку, потрібно помножити на кількість усіх S-блоків, яка повністю визначається кількістю раундів обох типів та їх структурою. Потім отриману величину потрібно розділити на величину r , що визначає довжину виходу на одній ітерації конструкції SPONGE.

5.5. Чисельні результати для MNT-сумісних параметрів

Таким чином, в попередньому пункті показано обчислення кількості раундів з повним шаром S-блоків для простого поля характеристики p, де бітова довжина характеристики дорівнює 753. У якості простого поля вибрали одне з полів, для яких визначені триплети MNT-4 та MNT-6. У раундах з повним шаром S-блоків будемо розміщати по 3 S-блоки, у раундах з частковим шаром – по одному S-блоку. Лінійний оператор A задамо MDS-матрицею розмірності 3x3. В цьому випадку ємність $c = 2 \cdot 752 = 1504$ та швидкість $r = 752$, якщо використовувати байтове подання. Степеневі S-блоки обирались з міркувань $s(x) = x^u \bmod p$, де $u = \min\{v \in N : (v, p-1) = 1\}$ =13. Для інверсних S-блоків та степеневих S-блоків виду $s(x) = x^{13} \bmod p$ кількість раундів з повним шаром S-блоків дорівнює 4 (2 раунди на початку, 2 у кінці, щоб виключити можливість як атак з вибраним відкритим текстом, так і атак з вибраним шифротекстом). Кількість раундів з частковим шаром S-блоків буде близько 60. В цьому випадку кількість констрейнтів на біт дорівнює 0.48 для степеневих S-блоків та 0.29 для інверсних S-блоків, що у 3.5-5.8 разів менше за аналогічний показник для функції Педерсена.

При цьому кількість раундів з повним шаром S-блоків визначалась з нерівності

$$\left(\frac{144}{p}\right)^{4r_{full}} \le 2^{-2N} = 2^{-6p},$$

що є достатньою кількістю і для степеневих, і для інверсних S-блоків. З нерівності отримали значення

$$r_{full} = \left\lceil \frac{6 \cdot 753}{4 \cdot 745} \right\rceil = 2,$$

тобто по два раунди з повним шаром S-блоків на початку алгоритму та у кінці.

Якщо ж, за рекомендацією авторів, додати по два додаткових раунди, то отримаємо (зберігаючи парність величини r_{full}): $r_{full} = 4$, тобто сумарна кількість таких раундів дорівнює 8.

Подсумки розділу

Отримані результати показали, що блоковий алгоритм шифрування HadesMiMC, що базується на операціях у простому скінченному полі, є стійким до лінійного та різницевого криптоаналізу, при правильному виборі параметрів цього алгоритму. Також, як наслідок, стійкою буде і геш-функція Poseidon, побудована з використанням конструкції SPONGE та перестановки HadesMiMC. Також було показано, як повинні бути вибрані параметри алгоритму, щоб гарантувати його заданий рівень стійкості. Крім того, обчислено кількість констрейнтів на біт інформації та показано, що алгоритм HadesMiMC та геш-функція Poseidon на сьогоднішній день не мають конкурентів при використанні у SNARK-доведеннях.

1. МОДЕЛІ ОСНОВНИХ АТАК НА БЛОКЧЕЙН ТА СПОСОБИ ПРОТИДІЇ ЦИМ АТАКАМ

Як було показано у Розділі 1, блокчейн є однією з найпривабливіших моделей децентралізованого середовища, яке можна застосовувати для збереження, обробки та пересилання інформації, забезпечуючи при цьому максимальний рівень її деперсоналізації. Також отримані результати свідчать про надзвичайну роль криптології для забезпечення "життєдіяльності та життєздатності" цього середовища. Дійсно, сам блокчейн, як технологія, базується на таких криптографічних механізмах, як цифровий підпис та функція гешування. Ці криптографічні перетворення можна вважати базовими, оскільки вони забезпечують коректне функціонування будь-якого блокчейну, незалежно від його специфіки, додаткових властивостей та призначення.

Отже, всі моделі порушника та всі атаки, направлені на зазначені криптографічні перетворення, залишаються загрозами для будь-якого блокчейну. Оскільки геш-функції та цифровий підпис використовуються у криптології вже не одне десятиріччя, то багато з таких моделей загроз є добре вивченими та дослідженими, при цьому описані та строго обґрунтовані необхідні та достатні умови успіху різних атак, розроблені методи оцінювання стійкості та побудови заздалегідь стійких криптографічних механізмів. З іншого боку використання геш-функції в протоколах консенсусу типу POW дещо змінює традиційну модель атак на криптографичні геш-функції як колізістійки зворотні перетворення, оскільки частина прообразу геш-коду є заздалегідь відомою.

В цьому розділі хочемо зосередитись на відносно нових атаках, які виникли тільки після появи блокчейну, та способах захисту від них. Ці атаки направлені саме на спосіб функціонування блокчейну і не є атаками на криптографічні механізми, які його обслуговують. Деякі з атак на блокчейн є загальними, тобто основну їх ідею можна застосовувати до будь-якого блокчейну, незалежно від протоколу консенсусу, який у ньому

використовується. Інші атаки є спеціальними атаками, тобто суттєво залежать саме від типу протоколу консенсусу.

У цьому розділі розглянемо дві моделі загальних атак на блокчейн, оскільки, на відміну від атак на різні криптографічні примітиви, вони досліджені набагато менше. Крім огляду та порівняльного аналізу цих атак, також будуть наведені основні способи протидії цим атакам, які суттєво використовують особливості функціонування та обслуговування блокчейну.

Тут і далі зберігаємо деяку "звичну" термінологію, яка належить до криптовалютного блокчейну, проте надамо цим термінам нового, більш широкого змісту. Наприклад, під "транзакцією" будемо розуміти не тільки (і не стільки) "справжню" криптовалютну транзакцію, а будь-яке перетворення інформації, яке виконується у блокчейні. Аналогічне зауваження стосується й деяких інших термінів, сутність яких, як правило, буде зрозумілою з контексту.

1.1. Дві моделі атак на блокчейн

На даний момент в літературі описується два типи атак на Біткойн, і обидві вони є саме атаками на блокчейн, надалі будемо їх називати "Атака з видимим Форком" (Атака I) і "Атака з невидимим Форком" (Атака II). Нижче буде наведено їх опис та порівняльний аналіз.

1.1.1. Атака I

Суть цієї атаки полягає в тому, що противник намагається створити якомога довший форк, намагаючись викладати свої блоки таким чином, щоб якомога далі підтримувати існування двох гілок однакової (або майже однакової) довжини. При цьому він не приховує самого факту побудови форку, і сторонній спостерігач легко може побачити, що більш довга гілка складається то з одних блоків, то з інших. Тобто деякі транзакції то

зникають, то з'являються. Чесні майнери, які теж беруть участь у побудові блокчейну, повинні дотримуватися протоколу майнінгу, тому їм доводиться продовжувати то один, то інший ланцюжок, таким чином мимоволі беручи участь в підтримці цього форку.

Робота, яка присвячена побудові оцінок стійкості блокчейну до цієї атаки, розглядає тільки випадок протоколу PoS. У цій роботі при певних припущеннях показано, що якщо частка зловмисника $\frac{m}{m+n}$ строго менше, ніж частка чесного $\frac{n}{m+n}$, то ймовірність форку експоненційно прагне до нуля, якщо довжина цього форку прямує до нескінченності. Іншими словами, для досить великих l ймовірність форку довжиною l має вигляд $e^{-f(l)}$, де $f(l)$ деяка позитивна функція. При цьому отриманий результат є виключно асимптотичним, тому неможливо сказати, яка довжина є "досить великою" для того, щоб можна було скористатися отриманими оцінками. Відповідно, отримані результати не дають відповіді на питання: при заданих m, n та (маленькому) $\varepsilon > 0$, якою повинна бути довжина l форка, щоб його ймовірність була менше цього заданого $\varepsilon > 0$.

Інша робота розглядає ту ж задачу, тобто ту ж модель атаки, але для протоколу PoW. У цій роботі для моделей Bitcoin і GHOST отримані наступні результати:

1) отримані аналітичні верхні оцінки ймовірності форку різної довжини l при заданих m, n й (маленькому) $\varepsilon > 0$;

2) за цими оцінками отримані чисельні результати для різних часток $\frac{m}{m+n}$ зловмисника.

Хоча отримані оцінки не виражаються безпосередньо через експоненційні функції від довжини входу, але тим не менше вони придатні для отримання чисельних результатів і побудови відповідних графіків. Причому графіки і демонструють експоненційне спадання ймовірності з

ростом довжини форку (при фіксованих m та n).

1.1.2. Атака II

Дана атака аналізується, щонайменше, в п’яти публікаціях, а також в незліченній кількості популярних статей в Інтернеті, і що дуже важливо - в курсі Прінстонського університету "Bitcoin and Cryptocurrency". В англомовних джерелах вона називається "Double Spending Attack", що можна перевести як "Атака подвійної витрати". Її суть в тому, що зловмисник намагається скористатися своєю монетою як "нерозмінним п'ятаком" з повісті "Понеділок, який починається в суботу" братів Стругацьких.

Технічно це відбувається так. Зловмисник в блоці з номером, наприклад, 5, виконує якусь транзакцію, переказуючи гроші постачальнику послуг або товарів за якусь покупку. Постачальник отримує ці гроші і, відповідно, поставляє покупцеві товар. Після отримання товару зловмисник швидко починає майнити інший блок з тим же номером 5, тобто блок, який слідує за блоком номер 4, але в якому або немає цієї транзакції, або він переводить ці гроші собі на якийсь іншу адресу. І щоб гарантувати прийняття чесними майнерами саме цього альтернативного ланцюжка, він на альтернативний блок з номером 5 намагається "причепити" якомога більше блоків. Якщо у нього вийде зробити альтернативний ланцюжок довшим, то саме він, відповідно до протоколу майнінгу, буде вважатися правильним. Очевидно, що чим більше частка зловмисника (не важливо, це обчислювальні потужності в разі PoW або частка стейка в разі PoS), тим більше у нього шансів виконати цю атаку. Зокрема, якщо частка зловмисника більше половини, то ймовірність успіху цієї атаки дорівнює 1.

Для захисту від цієї атаки Накамото в своїй найпершій роботі запропонував не надавати товар відразу, як тільки транзакція сталася, а почекати деякий час, точніше - кілька блоків після цієї транзакції, і тільки потім, якщо транзакція не зникла з блокчейну, надавати товар. В цьому

випадку зловмисник не може будувати форк відразу після оплати, оскільки тоді постачальник побачить, що транзакція то зникає, то з'являється в блокчейні, і відмовиться від угоди. Тому противник спочатку перечікує, поки на блок з транзакцією "наросте" потрібна кількість блоків підтвердження. В цей час він може нишком генерувати форк, який починається до блоку з транзакцією, тобто в наших позначеннях може генерувати альтернативний п'ятий блок і наступні за ним блоки, але ні в якому разі не викладати цей альтернативний ланцюжок, щоб постачальник не запідозрив недобре. Це перший етап атаки. А ось коли блоки підтвердження вже були сформовані і товар отримано, зловмисник намагається "наздогнати" існуючий ланцюжок, і це другий етап атаки. Припустимо, поки генерується 6 блоків підтвердження, противник зміг згенерувати 4 блоки альтернативного ланцюжка. І тепер він відстає принаймні на 2 блоки. Якщо коли-небудь в майбутньому він зможе згенерувати стільки блоків, щоб "наздогнати" існуючий ланцюжок, який, в свою чергу, теж буде весь цей час рости, то атака буде успішною. Зокрема, якщо йому вдалося згенерувати 7 або більше блоків уже на першому етапі атаки, поки він перечікував блоки підтвердження, то атака вже вдалася, наздоганяти нічого не потрібно. Після отримання товару він просто викладає свій, більш довгий ланцюжок, в якій гроші залишаються у нього ж.

Зауважимо, що в Атаки I немає таких двох етапів, там просто весь час тягнеться "явний" форк. При цьому, на відміну від Атаки I, вигода зловмисника в Атаки II очевидна: він і товар придбав, і гроші не витратив. А от у випадку Атаки I максимум, що він може зробити - це скомпрометувати криптовалюту, що не принесе йому ніякої вигоди; а якщо він теж володіє цією валютою, то навіть принесе збитки.

Тепер проаналізуємо результати різних авторів, в яких розглядається Атака II і оцінюється стійкість блокчейну до цієї атаки. Зауважимо, що у всіх перерахованих нижче роботах завдання ставиться таким чином: при заданих m та n, а також маленькому $\varepsilon > 0$, якою має бути кількість блоків підтвердження l після транзакції, щоб ймовірність успішної атаки була

менше цього самого заданого ε?

Як вже говорилося, перші результати по цьому завданню були отримані в роботі Накамото. Однак отримані вони були в припущеннях, які не зовсім відповідають реальній моделі. Це вже обговорювлось в розіділі 1.

У цьому розділі буде розглянуто параметри протоколу консенсусу, названого Proof-of-Accuracy (далі – PoA) – Доказ Точності.

1.2. Загальні ідеї протоколу PoA

Результати аналізу різних протоколів узгодження, заснованих як на принципах «proof-of-works», так і на інших, дають змогу сформулювати загальну постановку задачі створення ефективного за швидкодією та стійкого до централізації протоколу узгодження для технології блокчейн.

Таблиця 6.1. Аналіз протоколів узгодження

Тип протоколу узгодження	Аналіз алгоритмів визначення пріоритету учасника при генерації нового блоку
Proof-of-stake	Пріоритет учасника при генерації нового блоку у ланцюзі блоків залежить від розміру частки розподіленого цінного ресурсу, яким він володіє. За генерацію блоку учаснику збільшується його частка цінного ресурсу. Питання щодо того як розподілити частки між учасниками на етапі ініціалізації протоколу не розглядається.
Proof-of-Activity	Пріоритет учасника при генерації нового блоку у ланцюзі блоків залежить від його обчислювальних ресурсів, частки цінного ресурсу та «активності» в мережі. Чим більше частка та час знаходження в мережі тим більший пріоритет. Прикладом практичної реалізації певною мірою можна вважати криптовалюту DASH.
Proof-of-Burn .	Пріоритет учасника при генерації нового блоку у ланцюзі блоків залежить від розміру частки розподіленого цінного ресурсу, який був знищений учасником на попередньому етапі протоколу.
Proof-of-Capacity	Пріоритет учасника при генерації нового блоку у ланцюзі блоків залежить від розміру частки розподіленого цінного ресурсу, яким є місткість простору для зберігання даних.

Proof of Delegated Stake	Пріоритет учасника при генерації нового блоку у ланцюзі блоків формується в два етапи. На першому відбувається вибір підмножини учасників за результатами процедури голосування. Кількість голосів кожного учасника залежить від частки володіння цінним ресурсом. В другому етапі приймають участь тільки учасники, обрані за результатом першого етапу. Пріоритет учасника на другому етапі залежить від наявності учасника в мережі і його обчислювальних ресурсів. Застосовується у системах Bitshares та Steemit.

Нам потрібно розробити протокол узгодження, для якого ймовірність формування блоку чесним учасником набагато більша ніж імовірність створення блоку нечесним учасником протоколу $p >> q$, який залишається стійким до атаки централізації з плином часу, не вимагає значних обчислювальних ресурсів та є ефективним за швидкістю виконання транзакцій.

Перш ніж сформулювати основну ідею створення такого протоколу, зауважимо,що:

в усіх відомих протоколах узгодження ресурс, за яким визначається пріоритет учасника під час формування угоди, прямо чи опосередковано пов’язаний з деяким реальним цінним ресурсом; у протоколах типу «proof-of-works» це час, обчислювальні ресурси та електроенергія; у протоколах типу “proof-of-stake” це процент від загального цінного ресурсу, наприклад пам’яті («proof-of-capacity»), активності учасника («proof-of-activity»), репутації учасника («proof-of-signature») та ін.; цей зв’язок не дає змоги учасникам генерувати довільну кількість ресурсів і довільно підвищувати свій пріоритет;

учасники протоколу зацікавлені у проведенні транзакцій з метою додавання блоку за рахунок підвищення свого ресурсу, за яким визначається

пріоритет учасника протоколу;

у протоколах є механізм, який обмежує можливість підвищення пріоритету учасників певною межею для збереження децентралізації протоколу.

Сформулюємо основні ідеї створення нового протоколу узгодження.

За основу можна взяти гібрид протоколу типу «proof-of-works» та “proof-of-stake”.

Для обчислення ресурсу, який визначає рейтинг учасника протоколу в генерації нового блоку, потрібно мати не лише певний поріг обчислювальної складності задачі підвищення рейтингу, але й інформацію про вхідні дані (задані неповно і неточно), що дає змогу розв’язати задачу з потрібною точністю.

Для унеможливлення генерації довільної кількості ресурсів учасником протоколу до початку протоколу застосовується два типи обмеження: по-перше, регулюється чисельність учасників, які можуть взяти участь у протоколі; по-друге, окремі дані рейтингу учасників формуються тільки для поточного сеансу протоколу (як сеансові ключі у схемі Дифі-Гелмана), по-третє, інформація про дані, потрібні для обчислення рейтингу, розміщується на кількох ресурсах, за доступ до яких конкурують учасники протоколу угоди («свідома DDOS – атака») і пошук цих даних займає певний непокращувальний час (принцип «пошуку голки в стіжку сіна»). Через це потрібно розробити метод, який дає змогу забезпечити принципову неможливість розв’язання задачі (на якій засновано генерацію нового блоку) з потрібною точністю до певного часу. Практично станом на певний час не має існувати алгоритм розв’язання задачі з потрібною точністю.

Ці ідеї визначають такий підхід до побудови протоколу узгодження: пропонується змінити обчислення функції формування «цифрової пломби» (алгоритму додавання нового блоку в блокчейн у разі застосування протоколу угоди «proof-of-works») таким чином, щоб необхідні вхідні дані були задані неповно і неточно, а значення функції потрібно

обчислити з точністю, що задається деяким порогом. Інформація про вхідні дані розміщується на кількох ресурсах, за доступ до яких конкурують учасники протоколу угоди. Остання властивість дає змогу зрівняти шанси учасників протоколу з високопродуктивними і малопродуктивними обчислювальними ресурсами в боротьбі за право генерації нового блоку. Як теоретичну основу побудови та оцінювання стійкості протоколу узгодження на основі «доказу точності» пропонується обрати загальну теорію оптимальних алгоритмів, яка пов'язує існування і складність алгоритмів з точністю задання вхідних даних.

Можливо декілька варіантів реалізації протоколів узгодження з використанням зазначених вище ідей.

Варіант 1. Протокол «simple tickets»

Етап ініціалізації. За допомогою криптографічного протоколу спільної генерації випадкової [51] величини генерується випадкове число R . Число є невідомим для учасників протоколу. Від нього обчислюється та публікується геш-код $H(R)$. За допомогою (k, n) — порогового криптографічного протоколу розподілу секрету з досконалою стійкістю (наприклад, протоколу Шаміра) число R розміщується за різними випадковими адресам мережі (наприклад, вибраного пулу IP-адрес мережі Інтернет) таким чином, що деякі адреси не містять частин секрету.

Основний етап. Учасники протоколу намагаються зібрати k частин для відновлення секрету та за допомогою криптографічних протоколів цифрового нотаріату і доведення з нульовими знаннями довести факт відвідування IP-адрес та володіння числом R. Перемагає та генерує блок той, хто перший зібрав k частин для відновлення секрету. Зв’язок із реальними цінними ресурсами — це використання учасником протоколу реальних IP-адрес, кількість яких є бмеженою. Усі сеансові дані після генерації блоку анулюються як сеансові ключі у протоколі Діфі-Хелмана, тобто цінним ресурсом є реальна IP-адреса. Можливим ускладненням для порушників, які будуть намагатися використати велику кількість IP-адрес, може бути потреба

у розв’язанні деякої складної задачі під час реєстрації IP-адреси для участі в протоколі. Задачу можна сформулювати згідно з тим самим принципом виключення існування алгоритму (даних для її розв’язання) до початку протоколу узгодження.

Під час практичної реалізації зазначеного протоколу виникають окремі труднощі технічного характеру (наприклад, пов’язані з вибором оптимальної кількості IP-адрес, тощо), тому розглянемо інший варіант реалізації зазначених ідей.

Варіант 2. Протокол узгодження «proof-of-accuracy» або «доказ точності»

Інформація про вхідні дані, потрібні для розв’язання задачі генерації нового блоку розміщується на кількох ресурсах, за доступ до яких конкурують учасники протоколу угоди («свідома DDOS – атака»). Як примітиви для створення протоколу узгодження «Proof-of-accuracy» («протоколу доведення точності») (далі — «примітиви протоколу») використовують протокол сумісного генерування випадкового біта, протокол передачі із забуванням, протокол часового замку, протоколи доведення при нульових знаннях, протокол розподілу секрету, обчислення з максимально можливою точністю для заданої апріорної інформації.

Сформулюємо загальний опис протоколу в термінах визначених примітивів протоколу, а потім наведемо приклади реалізації з урахуванням сучасних можливостей хмарних технологій для проведення криптоаналізу [52].

Нехай $N(X)$ — інформація, потрібна для обчислення секрету в (k,n) - пороговій схемі розподілу секрету (будь-яка апріорна інформація про X), $S: X \times \Re_+ \to 2^G$ — оператор (в частковому вигляді функція) відновлення секрету, G — множина значень функції інформації про секрет (наприклад, як G можна використовувати вагу Хемінга, значення біта парності та ін.). Тоді інформаційним порогом складності обчислень секрету (максимально можливою точністю для заданої апріорної інформації, що характеризує

можливості виконання транзакції) є інформаційна неповнота N. Як $\Phi(N(X))$ обираємо множину ідеальних алгоритмів ϕ відновлення секрету. Тоді можна довести, що для будь-яких алгоритмів відновлення секрету, які реалізуються множиною $\Phi(N(X))$, існує $r(N(X)) \geq \varepsilon > 0$, де $r(N(X))$ — радіус інформації $N(X)$. Прикладом описаної задачі може бути задача обчислення коефіцієнтів многочлена за його значеннями у точках.

Розіб'ємо випадковим чином граф G блокчейну на підграфи G_i, з розподілом P^R, де R — випадкова величина, що визначає кількість вершин підграфів. Вершини підграфа генерують значення (для кожної задачі свої) випадкових величин Y_i з розподілами P^{Y_i}, які відповідають значенням многочлена в певних точках. Як тільки для якогось підграфа кількість інформації буде дорівнювати $r(N(X))$, відповідна підмножина учасників системи розв'язує задачу отримання всіх коефіцієнтів многочлена, продемонструвати яку вона може іншим учасникам мережі. Ймовірності p та q тут визначаються розподілами ймовірностей P^R та P^{Y_i}.

Етап ініціалізації. За допомогою примітиву протоколу «протокол сумісного генерування випадкового біта» (далі — «RBG») учасники протоколу генерують випадкові коефіцієнти многочлену, які за допомогою примітиву протоколу «передачі із забуванням» (далі — «OT») передаються одному випадково обраному спільнотою (також за допомогою RBG) тимчасовому координатору.

Усі дії користувачів протоколюються за допомогою механізму ЕЦП.

Етап наповнення. Тимчасовий координатор за допомогою довіреного генератора випадкових чисел генерує коєфіцієнти многочлена, які випадковим чином надає учасникам спільноти. Участь самого тимчасового координатора у протоколі узгодження виключається за допомогою криптографічного протоколу часового замку (далі — «TL»).

Етап здійснення транзакції та завершення узгодження. Перший із спільноти, хто збирає k з n частин секрету, доводить це тимчасовому

координатору та іншим за допомогою примітиву протоколу доведення при нульових знаннях (далі — «ZKNP»). Транзакція вважається узгодженою після таких перевірок та завершення участі у протоколу тимчасового координатора за допомогою примітиву TL.

Зауваження.

DOS-атака на одного з учасників спільноти не порушує протокол, адже ймовірність вибору його як «завершувача» транзакції є незначною.

DOS-атака на тимчасового координатора не порушує протокол, адже ймовірність вибору його як тимчасового координатора є незначною.

Для зменшення ймовірності нечесної поведінки тимчасового координатора застосовується довірений зовнішній генератор випадковості та додатковий параметр, пов'язаний із тимчасовим координатором, який застосовується у його примітиві TL.

1.3. Реалізація протоколу Proof-of-Accuracy

Для розуміння реалізації протоколу Proof-of-Accuracy зробимо декілька зауважень. Право генерації наступного блоку в протоколі консенсусу отримує учасник з найбільшим рейтингом S. Рейтинг кожного учасника побічно залежить від кількості цінного ресурсу, яким в протоколі є біла IP-адреса абонента. Для обчислення рейтингу S використовується лотерея. Винагорода за участь в лотереї не збільшує кількість цінних ресурсів. Лотерея реалізується за допомогою обчислюваною в момент часу t функції F (до моменту часу t функція F є необчислювальною через неповноту вхідної інформації). Для цього при обчисленні функції F використовується випадкова величина R в спільній генерації якій беруть участь не менше k учасників протоколу.

Протокол не ввідноситься ні до протоколів типу "proof-of-work" ні до протоколів типу "proof-of-stake". При обчисленні пріоритету учасника протоколу використовується частка цінного ресурсу і зібрана в сеансі

протоколу інформація (тобто при обчисленні наступного генеруючого блок інформація задана неповно, а обчислення проводяться з необхідною точністю). Додатково для захисту від атак «попереднього обчислення пріоритету» в сеансі бере участь випадкова інформація, генерація якої залежить від декількох учасників.

Цінним ресурсом є біла IP-адреса абонента. Звідси частка цінного ресурсу в учасників протоколу приблизно стабільна. Отже цінний ресурс (біла IP-адреса учасника) можна асоціювати зі ставкою (“stake”) в класичних “proof-of-stake” протоколах. Класичним рівнянням за яким обчислюється рейтинг в “proof-of-stake” протоколах є

$$S < \text{Stake} \cdot \text{Age} \cdot \text{target}$$

де S – межа ставки (рейтинг), необхідний для генерації блоку,

Stake - ставка кокретного учасника в даному сеансі протоколу,

Age - «вік ставок»,

target – змінна, яка необхідна для корекції складності генерації блоку.

На відміну від класичного рівняння в протоколі, що наводиться нижче, ставки не пов’язані із винагородою та лише непрямо пов’язані з рейтингом, необхідним для генерації наступного блоку.

Покроковий опис протоколу

Етап 1 Ініціалізація («розкидання каменів»)

Визначення поточної кількості та підмножини учасників протоколу. Спільна генерація та розподіл секрету між учасниками протоколу (між елементами множини B).

У лотереї беруть участь усі активні в даний момент часу учасники зі своїми IP-адресами, n учасників - $IP_1, \dots, IP_n$.

Для генерації i - того блоку в ланцюжку блоків вибираємо пороги складності $0 < l_i < 1$ и $k_i < n$. Вибір значення k_i визначається співвідношенням необхідної швидкості транзакцій і стійкості протоколу до атаки DSA. Вибором значення l_i визначається ймовірність обчислення

предиката, яка визначається кількістю спроб генерації нового блоку для кожного учасника протоколу.

Вибір k_i учасників протоколу (наприклад, за найменшими значеннями IP адрес або будь-якою іншою процедурою випадкового вибору учасників), які генерують випадкове число R_i за допомогою групового протоколу Діфі-Хелмана. Груповий протокол Діфі-Хелмана є розширенням протоколу Діфі-Гелмана для n учасників Довжина R_i повинна бути не менша за 1024 біт із міркувань стійкості до криптоаналізу.

Етап 2 Акуммуляція інформації (“збір каменів”)

Збір частин секрету учасниками протоколу з підмножини A (“друга лотерея”). Обчислення рейтингової функції учасниками протоколи із підмножини A.

Функцію $F_{l_i}: \{0,1\}^* \to \{0,1\}$ – визначаємо як предікат такий, що ймовірність $P(F_{l_i}(, R_i) = 1) = l_i$, де $P(X)$ – ймовірність випадкової величини X. Предікат побудований наступним чином:

Конкатенуємо: $X = H_{i-1} \vee IP_j \vee count_{1j} \vee R_i$, де H_{i-1} – геш-код попереднього блока ланцюжків блоків транзакцій, $count_{j1}, \dots, count_{ju}, \dots, count_{jm}$ – значення лічильника, перший індекс – порядковий номер абонента, другий – порядковий номер послідовності лічільника.

Обчислюємо геш-код $H(X)$, за допомогою геш-функції$H: \{0,1\}^* \to \{0,1\}^t$ (де tповинна бути не менша ніж 256 біт із міркувань стійкості).

Визначаємо парність $H(X)$ якщо значення $l_i = 1/2$ (оскільки значення $H(X)$ прямує до рівномірного розподілу). $F_{l_i}(H(X)) = 1$, якщо парність $H(X)$ дорівняє 1.

Для будь-якої $0 < l_i \approx \frac{C}{D} < \frac{1}{2}$ – визначаємо $H(X)\mathrm{mod}D$ (геш-код від конкатенації геш-коду попереднього блоку ланцюжку транзакцій, IP адреси, випадкової величини R_i та відповідного значення лічильника взятого за модулем D). Якщо $H(X)\mathrm{mod}D < C$, тоді $F_{l_i}(H(X)) = 1$, та $F_{l_i}(H(X)) = 0$ інакше. Для будь-якої $\frac{1}{2} < l_i \approx \frac{C}{D} < 1$ – визначаємо $H(X)\mathrm{mod}D$ (геш-код від конкатенації геш-коду попереднього блоку ланцюжку транзакцій, IP адреси, випадкової величини R_i та відповідного значення лічильника взятого за модулем D. Якщо $H(X)\mathrm{mod}D \geq C$ тоді $F_{l_i}(H(X)) = 1$ та $F_{l_i}(H(X)) = 0$ інакше. Тут C та D – деякі цілі числа.

Для i блоку кожен з n учасників обчислює дот значень $F_{l_i}(H_{i-1}, IP_j, count_{1j}, R_i)$, ... $F_{l_i}(H_{i-1}, IP_j, count_{mj}, R_i)$, де H_{i-1} – геш-код попереднього блоку ланцюжку тарнзакцій, $count_{j1}, \dots, count_{ju}, \dots, count_{jm}$ – відповідне значення лічильника.

Переможцем стає той, у кого для деякого $F_{l_i}(H_{i-1}, IP_j, count_{uj}, R_i) = 1$. Якщо для деякого $j_1 \neq j_2$ $F_{l_i}(H_{i-1}, IP_{j_1}, count_{j_1}, R_i) = F_{l_i}(H_{i-1}, IP_{j_2}, count_{j_2}, R_i)$, тоді переможцем стає той, у кого значення $count_{j_*}$ є меншим.

Переможець підписує блок транзакцій.

Етап 3. Верифікація

Мінімізація ймовірності "розгалужень" та атаки підміни блоку. Перевірка процесу генерації блоку учасниками протоколи із множини B.

Всі учасники перевіряють значення F_{l_i} та перевірка цифрового підпису блоку транзакцій та корекція параметрів k_i та l_i (при необхідності зміни ймовірності атак розгалужень та швидкості протоколу). Зменшення l_i зменшує ймовірність розгалужень, але збільшує час консенсусу і навпаки.

Чисельний приклад роботи протоколу

Етап 1.

Всі активні на даний час абоненти мережи с їх білими IP-адресами є учасниками протоколу та беруть участь у лотереї, n=5 учасників - IP_1, ... , IP_5, нехай$IP_1 = 78.27.168.220$, $IP_2 = 104.31.255.255$, $IP_3 = 27.121.104.0$, $IP_4 = 5.39.126.236$, $IP_5 = 5.135.53.40$.

Для генерації i – блоку ланцюжку блоків транзакцій в блокчейні, обираємо поріг складності $0 < l_i < 1$ и $k_i < n$. Нехай $k_i = 3$, let $l_i = 0{,}5$.

Інший приклад. Обираємо $k_i = 3$, let $l_i = 0{,}67 \cong \frac{2}{3}$, відповідно $C = 2$, $D = 3$.

Обираємо $k_i = 3$ учасників протоколу (наприклад, за найменшим значенням трьох найменших десяткових розрядів IP-адреси), зауважимо, що на практиці доцільно застосувати іншу процедуру випадкового вибору, наприклад обчисленням $H(IP_i \parallel random_i)$, де $random_i$- випадкове число, різне для i-того блоку та застосування до геш-кодів лексиграфічного порядку для забезпечення рівної участі всіх абонентів мережі у протоколі. В нашому випадку обираємо учасників з IP-адресами IP_3, IP_5, IP_1. Ми генеруємо випадкове число R_i із використанням пртоколу Діфі-Хелмана:

Обираємо $p = 23$, $q \mid (p-1) = 11$, $\alpha = 5$. Учасник з адресою IP_3 обирає випадкове число $r_1 = 7$, обчислює $PR_1 = 5^7 \bmod 23 = 17$ та надсилає PR_1до учасника з адресою IP_5. Учасник IP_5 обирає випадкове число$r_2 = 4$, обчислює $PR_2 = 5^4 \bmod 23 = 4$, $PR_{12} = 17^4 \bmod 23 = 8$ та надсилає $PR_1 = 17, PR_2 = 4, PR_{12} = 8$ до учасника IP_1. Учасник IP_1 обирає випадкове число $r_3 = 15$, обчислює $PR_{23} = PR_2^{r_3} = 4^{15} \bmod 23 = 3$, $PR_{13} = PR_1^{r_3} = 17^{15} \bmod 23 = 15$, $R_i = PR_{12}^{r3} = 8^{15} \bmod 23 = 2$ та надсилає $P_{23} = 3$ до учасника IP_3, а $P_{13} = 15$ до учасника IP_5. Учасник IP_3 обчислює $R_i = PR_{23}^{r1} = 3^7 \bmod 23 = 2$. Учасник IP_5 обчислює $R_i = PR_{13}^{r_2} = 15^4 \bmod 23 = 2$.

Інший приклад. Обираємо $p = 557$, $q \mid (p-1) = 139$, $\alpha = 2$. Учасник IP_3 обирає випадкове число $r_1 = 73$, обчислює $PR_1 = 2^{73} \bmod 557 = 162$ та надсилає $PR_1 = 162$ до учасника IP_5. Учасник IP_5 обирає випадкове число $r_2 = 21$,

обчислює $PR_2 = 2^{21} \mathrm{mod} 557 = 47$, $PR_{12} = PR_1^{r_2} \mathrm{mod} p = 162^{21} \mathrm{mod} 557 = 340$ та надсилає $PR_1 = 162, PR_2 = 47, PR_{12} = 340$ до учасника IP_1. Учасник IP_1 обирає випадкове число $r_3 = 12$, обчислює $PR_{23} = PR_2^{r_3} = 47^{12} \mathrm{mod} 557 = 547$, $PR_{13} = PR_1^{r3} = 162^{12} \mathrm{mod} 557 = 19$, $R_i = PR_{12}^{r3} = 340^{12} \mathrm{mod} 557 = 455$ та надсилає $P_{23} = 547$ до учасника IP_3, надсилає $P_{13} = 19$ до учасника IP_5. Учасник IP_3 обчислює $R_i = PR_{23}^{r1} = 547^{73} \mathrm{mod} 557 = 455$. Учасник IP_5 обчислює $R_i = PR_{13}^{r2} = 19^{21} \mathrm{mod} 557 = 455 = 0x1c7$.

Етап 2.

Функція $F_{\frac{1}{2}}: \{0,1\}^* \to \{0,1\}$ – предікат, визначений як $P(F_{\frac{1}{2}}(*, R_i) = 1) = \frac{1}{2}$, де $P(X)$ – ймовірність випадкової величини X. Предікат побудований так:

Нехай $H_{i-1} =$ 0x7be45e06e6993cd9bbe506594f3a09185953e60ee2f0a1ce80f7e2f8eb56c350 – для прикладу геш-код від повідомлення “Begin of blockchain chain.”, обчислений за SHA-256 алгоритмом.

Для учасника IP_3приймемо $count_{11} = 0x0000001$ X = 0x7be45e06e6993cd9bbe506594f3a09185953e60ee2f0a1ce80f7e2f8eb56c3502712110400000011c7 де $R_i = 455 = 0x1c7$. Обчислюємо геш-код

$H(X) =$ 0xa2d2746d63ebf75684effbec969996aa2813e5e683a9c72b5f9bffc45beed7bb за SHA-256 алгоритмом. Визначаємо парність $H(X)$ ($l_i = 1/2$). $F_{\frac{1}{2}}(H(X)) = 1$, оскільки значення $H(X)$ непарне. УчасникIP_3 стає переможцем! Він генерує цифровий підпис блоку транзакцій та надсилає X всім учасникам IP_1, IP_2, IP_4, IP_5.

Інший приклад з $l_i = 0{,}67 \cong \frac{2}{3}$, відповідно $C = 2$, $D = 3$.

Функція $F_{\frac{2}{3}}: \{0,1\}^* \to \{0,1\}$ – предікат, визначений як $P(F_{\frac{2}{3}}(*, R_i) = 1) = \frac{2}{3}$, де $P(X)$ – ймовірність випадкової величини X. Предікат побудований так:

Нехай$H_{i-1} =$ $0x7be45e06e6993cd9bbe506594f3a09185953e60ee2f0a1ce80f7e2f8eb56c350$ – для прикладу геш-код від повідомлення “Begin of blockchain chain.”, обчислений за SHA-256 алгоритмом.

Для учасника IP_3 приймемо$count_{11} = 0x0000001;$ $X =$ $0x7be45e06e6993cd9bbe506594f3a09185953e60ee2f0a1ce80f7e2f8eb56c35027121104000000011c7$ де $R_i = 455 = 0x1c7.$ Обчислюємо $H(X) =$ $0xa2d2746d63ebf75684effbec969996aa2813e5e683a9c72b5f9bffc45beed7bb$ за SHA-256 алгоритмом. Визначаємо $H(X) \bmod 3 = 2$ ($l_i = 2/3$). The $F_{\frac{2}{3}}(H(X)) = 0,$ because $H(X) \bmod 3 \geq 2.$

Для учасника IP_5приймемо $count_{11} = 0x0000001;$ $X =$ $0x7be45e06e6993cd9bbe506594f3a09185953e60ee2f0a1ce80f7e2f8eb56c35051355340000000011c7$ де $R_i = 455 = 0x1c7$. Обчислюємо геш-код

$H(X) =$ $0x0b3480025d29973c711ec925e109a39496369ab200de9f3732b8bba97c5dcc88$ за SHA-256 алгоритмом. Визначаємо $H(X) \bmod 3 = 1$ ($l_i = 2/3$). The $F_{\frac{2}{3}}(H(X)) = 1,$ оскільки $H(X) \bmod 3 < 2.$

Учасник IP_5 стає переможцем! Він генерує цифровий підпис блока транзакцій та надсилає X всім учасникам IP_1, IP_2, IP_3, IP_4.

Етап 3. Верифікація.

Всі учасники IP_1, IP_2, IP_4, IP_5 перевіряють$F_{\frac{1}{2}}$ учасника IP_3($F_{\frac{2}{3}}$ учасника IP_5) та верифіцирують цифровий підпис блока транзакцій.

1.4. Параметри протоколу консенсусу Proof-of-Accuracy

6.4.1 Постановка задачі

Вперше основні ідеї і положення цього протоколу були описані в роботі. Тут буде наведено і строго математично обґрунтовано вибір параметрів протоколу, при яких можна забезпечити його стійкість до різних атак на блокчейн. В першу чергу це стосується найпоширенішої атаки, яка в кріптовалютних блокчейнах називається атакою подвійної витрати (double spend attack). Оскільки розглядається більш широке застосування блокчейна, то аналог такої атаки будемо називати атакою підміни блока, що в точності відображає її сутність.

Зауважимо, що до атаки підміни блока схильні практично всі основні протоколи консенсусу. Однак строго обґрунтовані аналітичні результати для обчислення ймовірності цієї атаки на сьогоднішній день отримані тільки для найбільш "стародавнього" протоколу консенсусу – протоколу Proof-of-Work, або скорочено PoW, запропонованого в піонерській роботі Накамото. Коротко проаналізуємо роботи, в яких ці результати були отримані.

У своїй єдиній роботі Накамото першим привів оцінки ймовірності атаки подвійної витрати, проте його розрахунки були невірними. Вони були отримані в припущеннях, які не зовсім відповідають реальній моделі.

Перше припущення, що має місце в усіх наступних роботах по цій темі – це припущення про те, що момент генерації блока і момент його появи в мережі збігаються. Тобто час поширення блока всім вузлам мережі дорівнює нулю. Але якби це відповідало реальності, то ймовірність "ненавмисного" форку (відгалуження) дорівнювала б нулю, а насправді такі форки трапляються приблизно 6 раз на місяць. Правда, слід зауважити, що якщо час генерації блока набагато більше, ніж час поширення блока в мережі, то це припущення не буде сильно спотворювати отримані результати.

Друге припущення ще більш некоректне. Воно полягає в наступному: при обчисленні ймовірності атаки Накамото припускає, що якщо ймовірність деякої

події дорівнює p, то кількість випробувань до моменту появи n-го успіху в точності дорівнює $\frac{n}{p}$. З математичної точки зору це означає, що випадкову величину, що має від'ємний біноміальний розподіл, замінили її математичним сподіванням. З огляду на той факт, що дисперсія такої випадкової величини, взагалі кажучи, істотно відмінна від нуля, таке припущення вносить в результат відчутні похибки.

У цих спрощених припущеннях були отримані аналітичні вирази для ймовірності успіху атаки подвійної витрати, за якими можна отримувати чисельні значення при різних параметрах мережі. Експериментальні результати показали, що зі зростанням кількості блоків підтвердження ймовірність атаки убуває експоненційно, але аналітичних підтверджень такої експоненційної залежності немає.

У наступних роботах багато авторів намагалися позбутися другого припущення, проте отримати математично обґрунтовані результати вдалося вперше тільки в роботі. Однак перше припущення про миттєве поширення блока в мережі авторам обійти не вдалося. Некоректність цього припущення вперше була помічена в роботі, і в ній же автор (без доказів і обґрунтувань) привів якісь загальні формули, які нібито враховують час поширення блока. Однак у зв'язку з відсутністю доказів перевірити їх коректність не представляється можливим, а їх вигляд не дозволяє користуватися ними для отримання чисельних результатів.

Крім атаки підміни блоку, в літературі аналізується ще одна атака - атака розщеплення (Splitting Attack), або атака поділу мережі. Вона є більш потужною, але менш поширеною. Атака полягає в тому, що зловмисник, створивши форк, намагається розділити мережу, підтримуючи форк якомога довше за допомогою додавання нових блоків до коротшої гілки блокчейну. Опис такої атаки з'явився значно пізніше. Мета цієї атаки - створити в мережі умови, при яких консенсус неможливий протягом тривалого часу, що призведе до втрати довіри до криптовалюти і, відповідно, до падіння її вартості. При цьому,

якщо зловмисник раніше брав кредит в цій валюті, то віддавати цей кредит йому буде набагато простіше. На відміну від атаки підміни блоку, коли зловмисник потайки будує альтернативну гілку блокчейну, в атаці поділу мережі він відкрито бере участь в створенні форку і його підтримці. При сприятливих для нього умовах чесні майнери, які діють відповідно до протоколу консенсусу, теж будуть змушені мимоволі підтримувати форк, продовжуючи то одну, то іншу гілку. Оскільки тут в підтримці форку беруть участь не тільки ресурси зловмисника, а й, частково, чесних майнерів, то в ній легше домогтися успіху, ніж в атаці підміни блоку.

Побудова верхніх (не асимптотичних) оцінок ймовірності успіху для атаки розщеплення істотно складніше, ніж для атаки підміни блоку. Вперше верхні оцінки ймовірності успіху для обох атак, отримані для протоколу PoW без допущення про нульовий часу синхронізації. Слід зауважити, що в цій роботі розглянута модель з дискретним часом що не зовсім відповідає реальній ситуації. Разом з аналітичними оцінками були приведені числові результати, отримані за ці оцінками при різних параметрах мережі. Аналітичні і числові результати показали, що ситуація, при якій зловмисник синхронізований істотно краще, ніж чесні майнери, є вкрай небезпечною. У такій ситуації ймовірність атаки підміни блоку може дорівнювати одиниці навіть в тому випадку, коли обчислювальні потужності противника істотно менше, ніж у чесних майнерів. Тобто порушується основний принцип так званої "атаки 50%", заявлений в попередніх роботах - для гарантії успіху атаки супротивника потрібно мати 50% обчислювальних потужностей. Саме в цьому і криється небезпека так званої "часткової централізації" в майнінгу, тобто ситуації, коли майнери об'єднуються у великі майнінгові пули, особливо якщо учасники одного пулу територіально знаходяться поруч або мають якісь можливості для миттєвого пересилання блоку.

Виходячи зі сказаного вище, будемо враховувати час синхронізації мережі при обґрунтуванні вибору параметрів протоколу PoA. Більш того, розглянемо

саму загальну модель з безперервним часом: будемо припускати, що час синхронізації у чесних Майнер і у зловмисника різний. Зробимо кілька припущень на користь зловмисника, зокрема, допущення про його "централізації", що дає йому можливість практично миттєво пересилати блоки, і припущення про те, що він може певним чином впливати на час доставки блоків чесним майнерам.

Слід зауважити, що протокол PoA залишає менше можливості противнику для "часткової централізації", ніж протокол PoW. Тому одна з переваг цього протоколу в тому, що противник не може істотно зменшити свій час синхронізації, що зменшує його можливості атакувати при наявності невеликих обчислювальних потужностей.

Протокол PoA володіє також рядом інших переваг, зокрема він менш схільній до олігополії великих майнінгових пулів.

6.4.2. Позначення, визначення та припущення

У цьому параграфі введемо основні позначення і опишемо основні припущення даної моделі. Безліч чесних майнерів будемо позначати HMs (Honest Miners), а безліч зловмисників – MMs (Malisious Miners). Введемо наступні величини:

T_H – випадкова величина, що дорівнює часу, який HMs витрачають на створення одного блоку;

T_H' – випадкова величина, що дорівнює часу, який HMs витрачають на створення одного блоку і його поширення по всіх вузлах мережі;

T_M – випадкова величина, що дорівнює часу, який MMs витрачають на створення одного блоку;

T_M' – випадкова величина, що дорівнює часу, який HMs витрачають на створення одного блоку і його поширення по всіх вузлах мережі.

Як було показано в [53], випадкові величини T_H та T_M мають експоненійні розподіли:

$$F_{T_H}(t) = P(T_H < t) = 1 - e^{\alpha_H t},$$
$$F_{T_M}(t) = P(T_M < t) = 1 - e^{\alpha_M t}, \quad (6.1)$$

для деяких $\alpha_H > 0$, $\alpha_M > 0$. Фізичний сенс цих параметрів полягає в тому, що це середні інтенсивності генерації блоку для чесних майнерів і для зловмисника, відповідно.

Позначимо $\alpha = \alpha_H + \alpha_M$ загальну інтенсивність генерації блоків в мережі.

Модель роботи мережі, яку розглядаємо в цій роботі, буде базуватися на двох природних припущеннях.

Припущення 1. Будемо вважати, що час доставки блоку, створеного чесними майнерами, всім іншим майнерам (принаймні, чесним), не перевищує D_H. Більш того, зробимо припущення на користь зловмисника і будемо припускати, що він може впливати на доставку блоку іншим майнерам, наприклад, затримувати його на довільний час, що не перевершує D_H. Аналогічно, D_M – це або час доставки блоку, створеного нечесним майнером, іншим нечесним майнерам (якщо його вважаємо однаковим), або максимальний час поширення блоку серед нечесних майнерів.

Припущення 2. Зробимо ще одне припущення на користь зловмисника і вважаємо, що $0 \le D_M \le D_H$.

У наших позначеннях виконуються рівності:

$$T_H' = D_H + T_H,\ T_M' = D_M + T_M.$$

Позначимо p_H ймовірність того, що HMs створять наступний блок раніше, ніж його створять MMs, і $p_M = 1 - p_H$ – ймовірність протилежної події.

$$p_H = \frac{\alpha_H}{\alpha_H + \alpha_M},\ p_M = \frac{\alpha_M}{\alpha_H + \alpha_M}.$$

Величини p_H (p_M) будемо називати частиною загального хешрейта, яка належить HMs (MMS). У роботах співвідношення величин грає вирішальну роль при визначенні того, наскільки протокол вразливий до атаки підміни блоку. Однак в нашому випадку, з урахуванням ненульового часу синхронізації, ступінь вразливості до атаки буде визначатися іншими величинами.

Визначення 1. При заданих параметрах мережі α та $\Delta D = D_H - D_M$ будемо називати *границею безпеки* (security threshold) для протоколу консенсусу PoA найменше з таких значень p_0, що за умови $p_M \geq p_0$ ймовірність успіху атаки підміни блоку дорівнює 1, незалежно від кількості блоків підтвердження.

Зауважимо, що в моделі з нульовим часом синхронізації $p_0 = \frac{1}{2}$, звідки і пішла назва "атака 50%."

Позначимо p'_H ймовірність того, що HMs створять і поширять свій блок по всіх вузлах мережі (принаймні, по всім чесним вузлів) раніше, ніж це зроблять MMs; позначимо p'_M ймовірність альтернативної події, $p_M' = 1 - p'_H$.

Тоді

$$p'_H = P\left(T_H' < T_M'\right), \quad p'_M = P\left(T_M' < T_H'\right),$$

причому $p'_H + p_M' = 1$.

Величини є більш важливими при аналізі стійкості до атаки, ніж, оскільки вони враховують час синхронізації і більш реалістично відображають стан мережі. Далі буде показано, що ймовірність успіху атаки DSA залежить саме від величин. Зокрема, якщо величина D_H достатньо велика у порівнянні D_M, то для успішної атаки зловмисникові, взагалі кажучи, не обов'язково мати перевагу в хешрейті.

6.4.3. Визначення границі безпеки для протоколу PoA

У цьому пункті наведемо точні аналітичні вирази, що зв'язують, з одного боку, параметри мережі, такі як час синхронізації, інтенсивність виходу блоків і частку хешрейта зловмисника, а з іншого - кордон безпеки для цього протоколу. З використанням цих виразів, в наступному параграфі обґрунтуємо вибір параметрів мережі для протоколу консенсусу PoA.

Ми будемо використовувати базовий результат, який показує, що для гарантії успіху атаки супротивника зовсім не обов'язково володіти не менш ніж половиною всіх обчислювальних потужностей мережі. Тобто в більш адекватної

моделі, який розглядається в цій статті і враховує час синхронізації мережі, так звана "атака 50%" може перетворитися в "атаку 45%" або навіть в "атаку 30%", якщо зловмисник дуже добре синхронізований в порівнянні з чесними майнерами.

Як було показано раніше для протоколу консенсусу PoW в моделі, що враховує час синхронізації, визначальну роль будуть грати величини p'_H и p'_M. Саме ці величини визначають уразливість до атаки підміни блоку, і вони ж дозволяють встановити межу безпеки для протоколу консенсусу PoW.

Для протоколу PoA ці результати залишаються справедливими і їх доведення будуть аналогічними. Оскільки ці результати будуть використовуватися далі, наведемо їх тут без доказів.

Теорема 6.1. В наших визначеннях:

$$p'_M = 1 - e^{-\alpha_M \Delta D} \cdot \frac{\alpha_H}{\alpha_M + \alpha_H} = 1 - e^{-\alpha_M \Delta D} \cdot p_H;$$

$$p'_H = e^{-\alpha_M \Delta D} \cdot \frac{\alpha_H}{\alpha_M + \alpha_H} = e^{-\alpha_M \Delta D} \cdot p_H.$$

Теорема 6.2. Нехай α – інтенсивність генерації блоків, $\Delta D = D_H - D_M$. Тоді в заданій моделі протоколу POA межа безпеки може бути знайдена як рішення рівняння

$$2(1 - p_0) = e^{\alpha p_0 \Delta D}.$$

У подальшому викладі зробимо ще одне припущення на користь зловмисника. Будемо припускати, що він добре синхронізований, тобто $D_M = 0$. Зауважимо, що отримані в цьому припущенні аналітичні вирази для ймовірності атаки можна використовувати і тоді, коли, але в цьому випадку вони будуть верхніми оцінками ймовірності атаки, а не її точними значеннями.

В Таблиці наведені чисельні результати значень кордону безпеки для різних параметрів мережі.

Таблиця Значення межі безпеки при різних значеннях інтенсивності α генерації блоків (в блоках за секунду) і часу синхронізації D_H (в секундах).

α \ D_H	$D_H = 2$	$D_H = 5$	$D_H = 10$	$D_H = 20$	$D_H = 60$
$\frac{1}{600}$	0.49916736	0.49792102	0.49585079	0.49173685	0.47564318
$\frac{1}{60}$	0.49173685	0.47961146	0.46014582	0.42408032	0.31492306
$\frac{1}{6}$	0.42408032	0.33756934	0.24626202	0.15678438	0.06282608

6.4.4. Вибір параметрів протоколу консенсусу PoA і обґрунтування вибору

Додатково введемо такі позначення:

n - кількість вузлів мережі (або, що те ж саме, кількість IP-адрес);

m - кількість "корисних" вузлів мережі, тобто таких, які містять частини секрету;

k - кількість частин секрету, які потрібно зібрати для його відновлення;

d - час, необхідний для відвідування однієї IP-адреси.

Спочатку визначимо, як інтенсивність генерації блоків залежить від значень n, m, k та d.

Теорема 6.3. У наших позначеннях інтенсивність генерації блоків визначається наступною рівністю:

$$\alpha = \frac{n}{mkd}.$$

Доведення. Визначимо випадкову величину ξ на імовірнісному просторі, яке є безліччю IP-адрес, в такий спосіб:

$$\xi = \begin{cases} 1, \text{якщо IP – адреса містить інформацію,} \\ 0, \text{інакше.} \end{cases}$$

За визначенням, випадкова величина (6.6) має розподіл Бернуллі з

параметром $p = \frac{m}{n}$.

Визначимо випадкову величину ζ як кількість спроб в схемі Бернуллі до k-го успіху, тобто як кількість IP-адрес, які майнер повинен відвідати до того, як він збере потрібну кількість частин секрету. Тоді випадкова величина $\zeta - k$ дорівнює кількості невдач до k-го успіху, тому вона має від'ємний біноміальний розподіл ([30]) з параметрами (k, p) і математичним сподіванням $E(\zeta - k) = \frac{k(1-p)}{p}$. Тоді

$$E(\zeta) = \frac{k(1-p)}{p} + k = \frac{k}{p} = \frac{km}{n}.$$

Відповідно, середній час t (в секундах), необхідне для створення одного блоку, дорівнює $t = \frac{kmd}{n}$, а інтенсивність генерації блоків $\alpha = \frac{1}{t} = \frac{n}{kmd}$. Теорему доведено.

У мережі, що складається з $n = 1000$ вузлів, при $m = 950$, $k = 500$ та $d = 0.3$ секунди, отримуємо

$$\alpha = \frac{1000}{500 \cdot 950 \cdot 0.3} \approx 0.007 \text{ блоків за 1 секунду,}$$

тобто на генерацію одного блоку потрібно приблизно 142 секунди, або 2 хвилини 22 секунди.

Тепер визначимо загальні вимоги до роботи протоколу консенсусу, які і будуть накладати відповідні обмеження на його параметри. Потім окреслимо область їх значення більш точно. Після цього проаналізуємо, який буде межа безпеки протоколу, тобто яка частка ресурсу зловмисника є критичною для мережі.

Почнемо з того, що бажаємо зменшити до певного рівня "зайві витрати" - так звані orphan blocks, які виникають в тому випадку, корду два або болем блоків створені приблизно в один і той же час. В цьому випадку виникає розгалуження - форк (fork), що призводить до тимчасового поділу блокчейну, оскільки одна частина Майнер раніше побачить один блок, а інша частина - інший. Тому різні

Майнер будуватимуть блоки на різних гілках блокчейну. Наприклад, найбільший за довжиною форк для Біткойна за весь час його існування дорівнює шести блокам. Після того, як мережа знову синхронізується, одна з гілок втрачається, що призводить до втрати роботи, виконаної майнерами.

Тому параметри мережі повинні гарантувати "невелику" ймовірність розгалуження. Щоб зрозуміти, яка саме ймовірність є прийнятною, можна звернутися до мережі Біткойн, яка успішно функціонує вже понад 10 років.

Згідно зі статистичними дослідженнями, "зайві" блоки виникають в цій мережі приблизно 6 раз місяць. Тобто ймовірність виникнення такого блоку дорівнює

$$\frac{6}{6 \cdot 24 \cdot 30} \approx 1.4 \times 10^{-3}.$$

Ми візьмемо цю величину як базову, але округлимо її в меншу сторону, тобто будемо вимагати, щоб ймовірність форку не перевищувала 10^{-3} (для мережі Біткойн це відповідало б п'яти "зайвим" блокам на місяць).

Зробимо допущення на користь зловмисника і будемо припускати, що $\Delta_M = 0$, тобто $\Delta = \Delta_H$.

Теорема 6.4. Нехай задано довільне $\varepsilon > 0$. Для того, щоб ймовірність P_f форку була не більше, ніж ε, необхідно та достатньо, щоб виконувалася нерівність:

$$\alpha \cdot \Delta \le x_f,$$

де x_f є рішенням рівняння

$$1 - e^{-xf}\left(1 + x_f\right) = \varepsilon.$$

Доведення. Позначимо T випадкову величину, яка дорівнює часу до створення наступного блоку.

Відмітимо, що якщо якийсь B_1 створений, то час до створення наступного блоку B_2 не залежить від того, скільки часу було витрачено на створення попереднього блоку.

Тому випадкова величина T має експоненційний розподіл з параметром α:

$$P(T \le t) = 1 - e^{-\alpha t}.$$

Ненавмисний форк може виникнути тоді і тільки тоді, якщо за час, який не перевищує час Δ синхронізації мережі, два або більше майнерів створили блоки.

Оскільки час генерації блоків має експоненціальне розподіл, то випадкова величина η_t, що дорівнює кількості блоків, створених за час t, матиме розподіл Пуассона з параметром $\alpha \cdot t$:

$$P(\eta_t = k) = e^{-\alpha t} \frac{(\alpha t)^k}{k!},\ k = 0,1,2,...$$

Тому ймовірність форку дорівнює

$$P_f = P(\eta_\Delta \ge 2) = 1 - P(\eta_\Delta = 0) - P(\eta_\Delta = 1) = 1 - e^{-\alpha\Delta} - \alpha\Delta e^{-\alpha\Delta} = 1 - e^{-\alpha\Delta}(1 + \alpha\Delta).$$

Для завершення докази нам потрібно показати, що функція $P_f(x) = 1 - e^{-x}(1 + x)$ є зростаючою. Дійсно, її похідна

$$P_f'(x) = e^{-x}(1 + x) - e^{-x} = xe^{-x}$$

невід'ємна при $x \ge 0$ та позитивна при $x > 0$.

Тому якщо для деякого x_f виконується рівність

$$P_f(x_f) = \varepsilon,$$

то при $x \in (0, x_f)$ буде виконуватися нерівність $P_f(x_f) < \varepsilon$, й теорему доведено.

Обчислимо x_f з умови $P_f(x_f) = 10^{-3}$ та отримаємо нерівність

$$\alpha\Delta \le 0.05,$$

яке і будемо використовувати для визначення параметрів n, m, k мережі.

З теореми 3 й нерівності (3.7) отримаємо

$$\frac{m}{nkd} \le 0.05.$$

Зауважимо, що кількість користувачів мережі як правило $n \ge 10^4$. Візьмемо цю величину як базову і покажемо, як визначаються інші параметри.

Оскільки ми, взагалі кажучи, не можемо впливати на параметри d та Δ, а параметр n повинен мати можливість зростати без особливих обмежень, то обмеження можемо накладати тільки на параметри m та k. Також варто

відзначити, що параметр не повинен бути істотно меншим, ніж n, щоб ймовірність успіху у відповідній схемі Бернуллі була близькою до 1. Експериментальні результати показали, що найкраще вибирати $m \approx 0.95n$. Звідси отримуємо:

$$\frac{0.95 \cdot n \cdot \Delta}{k \cdot n \cdot d} \le 0.05, \text{ або } k \ge \frac{0.95 \cdot n \cdot \Delta}{0.05 \cdot n \cdot d} = \frac{19 \cdot \Delta}{d}.$$

Однак, за визначенням, $k \le m$, тому маємо подвійну нерівність:

$$\frac{19\Delta}{d} < k < 0.95n.$$

Обчислимо нижню межу значень, які може приймати параметр k при різних, найбільш часто зустрічаються значеннях d та Δ. Значення нижньої межі наведені в Таблиці 6.2.

Таблиця 6.3. Нижня границя значення параметра k

Δ / d	5c	10c	20c	30c	60c
0.3c	316.7	633	1267	1900	3800
0.5c	190	380	760	1140	2280
1c	95	190	380	570	1140
2c	47.5	95	190	285	570
3c	32	63	127	190	380
5c	19	38	76	114	228

Проаналізуємо, як змінюється межа безпеки при зміні параметра k. Згідно з визначенням, межа безпеки p_0 є найменший часткою зловмисника, яка дозволять йому атакувати блокчейн з ймовірністю успіху 1. Як було доведено в теоремі 6.2, межа безпеки визначається як розв'язок рівняння $2\left(1-x\right) = e^{\alpha \cdot \Delta \cdot x}$.

Покажемо, що межа безпеки зменшується з ростом добутку $\alpha \cdot \Delta$. Для цього доведемо наступну теорему.

*Теорема 6.5.*Нехай неявна функція задана формулою

$$F\left(x, y\right) = 2\left(1-x\right) - e^{xy} = 0.$$

Тоді існує функція $x = x\left(y\right)$, $y \ge 0$, яка приймає значення в інтервалі $\left(0,1\right)$, така, що для неї виконується рівність:

$$F\left(x\left(y\right),y\right)=0.$$

При цьому функція $x=x\left(y\right)$ є спадаючою.

Доведення. Для доведення досить показати, що функція $y=y\left(x\right)$ є безперервною і спадаючою. Тоді вона буде біекцією, отже, у неї буде існувати зворотна функція, яка теж буде безперервною і спадаючою.

З (6.10) отримуємо:

$$y\left(x\right)=\frac{\ln 2\left(1-x\right)}{x}.$$

Покажемо, що похідна цієї функції негативна, тобто що

$$y'\left(x\right)=\frac{\frac{x}{x-1}-\ln 2\left(1-x\right)}{x^2}<0.$$

Для цього достатньо довести, що мислитель в (6.11) менше нуля: $\frac{x}{x-1}-\ln 2\left(1-x\right)<0$, або що

$$x-\left(x-1\right)\ln 2\left(1-x\right)>0,$$

оскільки $x-1<0$.

Позначимо

$$f\left(x\right)=x-\left(x-1\right)\ln 2\left(1-x\right).$$

Тоді

$$f'\left(x\right)=1-\ln 2\left(1-x\right)-1=-\ln 2\left(1-x\right),$$

звідки $f'\left(x\right)=0\Leftrightarrow x=\frac{1}{2}$, причому з (3.13) видно, що $x=\frac{1}{2}$ є точкою мінімуму.

Обчислимо значення функції $f\left(x\right)$ в точці мінімуму:

$$f=\frac{1}{2}+\frac{1}{2}\ln\left(2\cdot\frac{1}{2}\right)=\frac{1}{2}>0.$$

Крім того,

$$\lim_{x\to 0+0}f\left(x\right)=f\left(0\right)=\ln 2>\frac{1}{2};$$

$$\lim_{x \to 1+0} f(x) = 1 > \frac{1}{2},$$

тобто в точці $x = \frac{1}{2}$ функція $f(x)$ приймає найменше значення, і воно є додатним. Тому $f(x) \geq \frac{1}{2} > 0$, й (6.12) виконується, а, отже, виконується і (6.11), тобто $y'(x) < 0$. Отже, функція $y(x)$ є спадаючою. Вона також є неперервною, як композиція неперервних функцій. Тому обернена функція $x = x(y)$ існує і є спадною. Теорему доведено.

Наслідок 1. Функція $p_0 = p_0(\alpha \cdot \Delta)$ є спадаючою, тобто границя безпеки p_0 спадає із зростанням добутку $\alpha \cdot \Delta$.

Проаналізуємо, яка межа безпеки відповідає обраному нами значенням $\alpha \cdot \Delta = 0.05$. Для цього нам потрібно обчислити таке значення, яке є рішенням рівняння $2(1 - p_0) = e^{0.05 p_0}$.

Рішення цього рівняння приблизно дорівнює $p_0 \approx 0.488$, в той час як максимально можливим значенням границі безпеки (в ідеальній для чесних майнерів ситуації) є $p_0 = 0.5$. Тобто при заданих обмеженнях $\alpha\Delta < 0.05$ значення межі безпеки буде менше ідеального всього на 1.2%, що можна вважати незначним пониженням безпеки. Іншими словами, мережа буде стійкою до тих пір, поки частка зловмисника не перевищує 48.8%.

Підкреслимо, що теоретично досяжна межа безпеки - 50% - можлива лише за умови нульового часу синхронізації для чесних майнерів, що є недосяжним на практиці.

Нерівність (6.9) дає нам досить широкі межі для параметра k. З практичних міркувань, які були перевірені експериментами, зручно вибирати значення k з співвідношення

$$k \approx \frac{1}{2} n.$$

Таке значення задовольняє нерівності за умови, що $n \geq 10000$ та $\Delta < 263d$, що майже завжди виконується на практиці.

Розрахуємо інтенсивність виходу блоків при деяких значеннях параметрів, які задовольняють заданим обмеженням. Нехай $n=10^4$, $m=0.95n=9500$, $k=0.5n=5000$, а параметр d нехай приймає змінні значення. Результати обчислень наведені в таблиці 6.3.

Таблиця Значення інтенсивності α при $n=10^4$, $m=9500$, $k=5000$ та змінному d.

d	$5\cdot10^{-3}$	10^{-2}	$2\cdot10^{-2}$	$3\cdot10^{-2}$	$5\cdot10^{-2}$	0.1	0.2
α	$3.8\cdot10^{-2}$	$19\cdot10^{-3}$	$9.5\cdot10^{-3}$	$6.3\cdot10^{-3}$	$3.8\cdot10^{-3}$	$1.9\cdot10^{-3}$	$0.95\cdot10^{-3}$

Всі значення параметра в цій таблиці, крім останнього, більше, ніж значення відповідного параметра для мережі Біткойн (який дорівнює $1.6\cdot10^{-3}$). Відмітимо, що параметр $k=0.5n=5000$ забезпечує "маленьку" ймовірність форку лиш за умови $\Delta\leq 263d$. Наприклад, якщо $d=0.1$, то такий вибір параметра k можливий лише при $\Delta\leq 26.3$. Якщо ж це нерівність не виконується, то потрібно звелічіть параметр k, що, відповідно, призведе до зменшення інтенсивності виходу блоків. Але це об'єктивна проблема, яка властива будь-якому блокчейну, незалежно від протоколу консенсусу.

Однією з переваг даного протоколу є те, що змінюючи значення параметра k можемо адаптувати інтенсивність виходу блоків в залежності від середнього часу синхронізації мережі. Наприклад, якщо воно гарантовано "маленьке", то відповідно можна істотно підвищити інтенсивність генерації блоків, і навпаки. Аналогічна адаптація можлива і по відношенню до параметру d.

Підсумки до розділу

1. У даному розділі розглядається новий протокол консенсусу на блокчейні. Він є адаптивним, тобто дає можливість вибирати його параметри в залежності від існуючих параметрів мережі, на які не можемо впливати. Вибір параметрів протоколу виконується виходячи з таких міркувань:

- ймовірність форку є не більшою, ніж для мережі Біткойн;
- інтенсивність виходу блоків може бути більше, ніж в мережі Біткойн;
- стійкість мережі бути практично така ж, як для "ідеальної" мережі.

2. Побудова протоколів узгодження на основі принципів, що поєднують переваги протоколів «доказу роботи» та «доказу частки (володіння, активності та ін.)» вважається перспективною з погляду економії обчислювальних ресурсів та збереження децентралізації. Протокол PoA має ефективність за часом на рівні протоколів, що використовують «візантійські угоди», але має менш суворі вимоги до кількості нечесних учасників протоколу. Якщо цей протокол консенсусу використовується в якійсь локальній мережі, для якої час синхронізації гарантовано маленький, то інтенсивність генерації блока можна збільшувати в багато разів.

3. Для "неідеальної" мережі неможливо неконтрольовано збільшувати інтенсивність генерації блока. Але це обмеження не є особливістю саме цього протоколу, а є загальним недоліком будь-якого протоколу консенсусу для блокчейну.

4. Щоб мати можливість суттєво збільшити інтенсивність генерації блоків, потрібно від блокчейну переходити до іншої структурі. Публікації останніх років пропонують використовувати замість блокчейну такі структури, як DAG. Але практично у всіх відомих роботах з цього напрямку гарантія досягнення консенсусу і обгрунтування стійкості запропонованої структури до різних атак, а також формулювання і обґрунтування аналогів інших необхідних властивості, притаманних блокчейну, або не є строго математично обґрунтованими, або містять помилки в доказах, або мають лише асимптотичні оцінки.