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

«На правах рукопису»
УДК 004.056.53

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

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-науковою програмою
«Математичні методи криптографічного захисту інформації»

зі спеціальності: 113 Прикладна математика
на тему: **«Побудова та узагальнення моделі стійкої до ASIC-атак хеш-функції Verthash»**

Виконав:
студент II курсу, групи ФІ-42мн
Молдован Дмитро Володимирович _____

Керівник:
доцент кафедри ММЗІ, к.т.н., доцент
Кучинська Наталія Вікторівна _____

Рецензент:
Заст. дир. ФТІ з наукової роботи, к.ф-м.н., доцент
Терещенко Іван Миколайович _____

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

Київ — 2026

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

Рівень вищої освіти — другий (магістерський)
Спеціальність — 113 Прикладна математика,
ОНП «Математичні методи криптографічного захисту інформації»

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

ЗАВДАННЯ
на магістерську дисертацію

Студент: Молдован Дмитро Володимирович

1. Тема роботи: «*Побудова та узагальнення моделі стійкої до ASIC-атак хеш-функції Verthash*», науковий керівник дисертації: доцент кафедри ММЗІ, к.т.н., доцент Кучинська Наталія Вікторівна,

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

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

3. Об'єкт дослідження: *процес функціонування моделі стійкої до ASIC атак хеш-функції Verthash*

4. Предмет дослідження: *модель стійкої до ASIC-атак хеш-функції Verthash*

5. Перелік завдань:

1) аналіз опублікованих джерел, що розкривають принципи роботи алгоритму хешування Verthash, ASIC-схем, майнінгу криптовалют;

2) аналіз математичного апарату алгоритму хешування Verthash та його функціоналу;

3) побудова формалізованого математичного опису моделі алгоритму хешування Verthash;

4) дослідження основних криптографічних характеристик побудованої моделі та аналіз потенційних напрямків масштабування та узагальнення моделі для подібних ASIC-стійких систем.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: Презентація доповіді

7. Орієнтовний перелік публікацій: Доповідь на XXIV Всеукраїнській науково-практичної конференції студентів, аспірантів та молодих вчених у секції «Актуальні проблеми криптографічного захисту інформації»

8. Дата видачі завдання: 10 вересня 2025 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	01-15 вересня 2025 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	Вересень-жовтень 2025 р.	Виконано
2	Ознайомлення з особливостями практичної реалізації моделі стійкої до ASIC-атак схеми хешування Verthash.	Жовтень 2025 р.	Виконано
3	Формалізація підходів до аналізу схем хешування типу Hashimoto, Ethash, Vertcoin	Листопад 2025 р.	Виконано
4	Побудова формалізованого математичного опису моделі алгоритму хешування Verthash	Грудень-січень 2025 -2026 рр.	Виконано
5	Дослідження основних криптографічних характеристик побудованої моделі	Лютий 2026 р.	Виконано
6	Аналіз потенційних напрямків масштабування та узагальнення моделі для подібних ASIC-стійких систем	Березень 2026 р.	Виконано
7	Редагування, узагальнення та оформлення результатів проведеного дослідження	Квітень 2026 р.	Виконано
8	Виступ на XXIII Всеукраїнській науково-практичної конференції студентів, аспірантів та молодих вчених	Травень 2026 р.	Виконано

Студент _____ Дмитро МОЛДОВАН

Керівник _____ Наталія КУЧИНСЬКА

РЕФЕРАТ

Кваліфікаційна робота містить: 51 стор., 4 рисунки, 2 таблиці, 11 джерел.

Існуючі ASIC-стійкі алгоритми хешування, що застосовуються в блокчейн-мережах на основі протоколів консенсусу Proof of Work функціонують на основі відкритих специфікацій, проте потребують теоретичного узагальнення, через брак формального математичного опису їх структурних схем. Розробка математичних моделей алгоритмів хешування, направлених на протидію атакам на спеціалізованому обладнанню (ASIC), є основою до аналізу їх стійкості та побудови нових більш стійких децентралізованих систем. З огляду на це, метою кваліфікаційної роботи є формалізація, структурно-графічне моделювання та аналіз факторів стійкості до ASIC алгоритму хешування Verthash.

Серед головних результатів, отриманих в ході виконання дослідження, є формалізація та представлення у вигляді графічних схем архітектури алгоритму хешування алгоритму Verthash. Розроблено систематичну модель генерації базового масиву псевдовипадкових даних `verthash.dat`. Систематизовано технічні чинники забезпечення стійкості до атак на спеціалізованому обладнанню (ASIC), що базуються на обмеженні пропускної здатності шини пам'яті. Проведено порівняльний аналіз технічних та криптографічних характеристик пам'яттєво-залежних алгоритмів хешування (Dagger-Hashimoto, Ethash, Verthash), результати проведеного аналізу представлено у вигляді аналітичної таблиці.

БЛОКЧЕЙН, PROOF OF WORK, ХЕШУВАННЯ, VERTHASH, ASIC

ABSTRACT

The qualification work contains: 51 pages, 4 figures, 2 tables, 11 sources.

Existing ASIC-resistant hashing algorithms used in blockchain networks based on Proof of Work consensus protocols operate on the basis of open specifications, but require theoretical generalization due to the lack of a formal mathematical description of their structural schemes. The development of mathematical models of hashing algorithms aimed at counteracting attacks on specialized equipment (ASIC) is the basis for analyzing their stability and building new, more stable decentralized systems. Considering this, the purpose of the qualification work is the formalization, structural and graphical modeling and analysis of factors of ASIC-resistance of the Verthash hashing algorithm.

Among the main results obtained during the research, there is a formalization and presentation in the form of graphic diagrams of the architecture of the hashing algorithm Verthash. A systematic model for generating the base array of pseudorandom data `verthash.dat` has been developed. The technical factors ensuring resistance to attacks using specialized hardware (ASICs), based on limiting memory bandwidth, have been systematized. A comparative analysis of the technical and cryptographic characteristics of memory-dependent hashing algorithms (Dagger-Hashimoto, Ethash, Verthash) has been carried out, the results of the analysis are presented in the form of an analytical table.

BLOCKCHAIN, PROOF OF WORK, HASHING, VERTHASH, ASIC

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Bitcoin — перша успішна практична реалізація концепту криптовалюти - альтернативного виду цифрових грошей, безпека якої базується на криптографічних властивостях.

Ethereum — криптовалюта, у мережі якої набув широкого практичного застосування алгоритм Dagger-Hashimoto (Ethash) з підвищеною ASIC-стійкістю.

Vertcoin — криптовалюта, В мережі якої було використано ASIC-стійкий алгоритм хешування, побудований на основі Ethash - Verthash

PoW — (Proof of Work) алгоритм консенсусу в блокчейні, який передбачає виконання обчислювальної роботи для підтвердження транзакцій та створення нових блоків.

ASIC — (Application-Specific Integrated Circuit) спеціалізована інтегральна схема, призначена для ефективного виконання конкретного обчислювального завдання, зокрема майнінгу криптовалют.

FPGA — (Field-Programmable Gate Array) програмована логічна інтегральна схема, конфігурація якої може бути змінена після виготовлення для реалізації спеціалізованих обчислювальних алгоритмів.

ВСТУП

Актуальність дослідження. За останні два десятиліття криптовалюти пройшли шлях від теоретичного концепту «на папері» до зручного фінансового інструменту, який породив власну екосистему з великою капіталізацією, та який все більше інтегрується в життя людей на побутовому рівні. Створення такого нового концепту цифрових грошей було зумовлене проблемами які не могли бути вирішені традиційними платіжними методами.

Першою криптовалютою, яка стала втіленням цих ідей став Bitcoin. За задумом творця, система би існувала у вигляді багатьох окремих вузлів (комп'ютерів), що можуть надсилати один одному транзакції та брати участь у математичних обчисленнях, необхідних для побудови блокчейну - механізму, який виконує функції журналу транзакцій та підтверджує їхню автентичність. Отримана таким чином децентралізована фінансова система не потребувала б наявності третьої сторони для функціонування, а отже унеможливлювала би накопичення повноважень у певного регулятора, який би мав змогу керувати транзакціями, вводити мита, впливати на учасників та фінансову систему в цілому.

Але на практиці, дана система не відповідає основним принципам, які оголошували розробники першої криптовалюти. На даний час, усі обчислення, необхідні для побудови блокчейну для Bitcoin проводяться великими майнінг фермами, що являють собою обчислювальні дата центри з великою кількістю потужних спеціально розроблених та з'єднаних між собою в один пул апаратних пристроїв (ASIC). Тобто на практиці така система знову залежить від обмеженого кола користувачів, що уможливлюють функціонування всієї системи.

Саме тому була запропонована криптовалюта Vertcoin, яка у своїй внутрішній структурі базується на принципово відмінному механізмі.

Головним в такому механізмі є алгоритм хешування Verthash, який вимагає багаторазового доступу до пам'яті, в той час як майнінг Bitcoin напряму залежить від обчислювальної потужності задіяних апаратних пристроїв.

Актуальність дослідження зумовлена недостатністю висвітлення питань, що стосуються дослідження стійкості моделей хеш-функцій до ASIC-атак.

Мета дослідження. Метою даної роботи є дослідження моделі стійкої до ASIC-атак хеш-функції Verthash, систематизація відомостей про ефективність її роботи. Окрім цього, важливою є перевірка відповідності заявлених особливостей роботи Vertcoin, зокрема хеш-функції Verthash, її основним принципам з практичної точки зору.

Серед основних *завдань*, які були сформульовані та виконані в ході виконання дослідження, були:

1) аналіз опублікованих джерел, що розкривають принципи роботи алгоритму хешування Verthash, ASIC-схем, майнінгу криптовалют;

2) аналіз математичного апарату алгоритму хешування Verthash та його функціоналу;

3) побудова формалізованого математичного опису моделі алгоритму хешування Verthash;

4) дослідження основних криптографічних характеристик побудованої моделі та аналіз потенційних напрямків масштабування та узагальнення моделі для подібних ASIC-стійких систем.

Об'єктом дослідження є процес функціонування моделі стійкої до ASIC-атак хеш-функції Verthash.

Предметом дослідження є модель стійкої до ASIC-атак хеш-функції Verthash.

При розв'язанні поставлених завдань були використані такі *методи дослідження*: методи комп'ютерного та математичного моделювання

Наукова новизна отриманих результатів полягає у тому що було побудовано математичну модель алгоритму хешування стійкої до ASIC-

атак хеш-функції Verthash та досліджено її безпекові властивості.

Практичне значення результатів полягає у формальному описі математичної моделі алгоритму хешування стійкої до ASIC-атак хеш-функції Verthash, який розширює теоретичну базу для подальших досліджень ASIC-стійких хеш-функцій.

Апробація результатів та публікації. Доповідь на XXIV Всеукраїнській науково-практичної конференції студентів, аспірантів та молодих вчених у секції «Актуальні проблеми криптографічного захисту інформації»

1 АНАЛІЗ ОПУБЛІКОВАНИХ ДЖЕРЕЛ ЩОДО АЛГОРИТМУ ХЕШУВАННЯ VERTHASH, ASIC-СХЕМ, МАЙНІНГУ КРИПТОВАЛЮТ

У межах даного розділу розкрито особливості будови та функціонування криптовалют Bitcoin, Ethereum та Vertcoin. Виходячи з філософії та умов, що передували виникненню цих систем, було виділено основні відмінності у роботі їхніх протоколів консенсусу PoW. Додатково, для розуміння усіх технічних процесів, що являють собою функціонал даних цифрових валют, особливу увагу було приділено питанням практичної реалізації їх роботи за допомогою ASIC-схем, FPGA-схем та інших апаратно-технічних рішень.

1.1 Функціонування екосистеми Bitcoin

Першою робочою реалізацією концепту валюти, безпека якої базувалась би на криптографічних властивостях є монета Bitcoin. Необхідно розуміти головні проблеми теперішньої грошової системи, виклик яким і був кинутий розробниками першої криптовалюти, щоб далі перейти до опису проблем, які виникли з функціонуванням її власної схеми та їх рішень, представлених альтернативними розробками, такими як, наприклад, Etherium та Vertcoin.

1.1.1 Створення та принципи роботи мережі Bitcoin

Сплеск інтересу до терміну криптовалют почався за 10 років до створення Bitcoin, ще у 1998 році з публікації Wei Dai - b-money [4]. Дана публікація частково і стала поштовхом до першої успішної практичної

реалізації концепту цифрових грошей у 2008 році [9]. Якщо розглянути поняття грошей глобально, то ними може бути будь-який об'єкт, що може виступати у вигляді плати за речі чи послуги, або який має цінність з економічної точки зору для держави чи будь-якого іншого соціально-економічного утворення. Схематично функціонування монети Bitcoin спирається на ідею використання криптографії для створення нових монет та проведення транзакцій, в противагу моделям, що опираються на довіру до центральних банків. Зокрема, наявність такої третьої сторони необхідна під час проведення транзакцій онлайн, щоб унеможливити атаку подвійної витрати [9, 1, с. 1].

У фінансовій системі світу, що існує зараз, люди мають довіряти центральним банкам, які оголошують цінність грошам (курси валют), правила користування та розпорядження ними. Зберігаючи свої гроші в банках, люди мають довіритись, що одного дня ці гроші просто не зникнуть. Функціонування такої системи є досить дорогим та складним процесом.

Також, гроші, що перебувають на рахунках людей в банку насправді ніколи не лежать та не очікують коли користувачі приймуть рішення отримати їх назад. Банки використовують ці гроші для своєї діяльності, наприклад для видачі кредитів, інвестицій, міжбанківських операцій тощо. Усі гроші в банку постійно перебувають в обігу і якщо раптом протягом короткого проміжку часу усі користувачі захочуть вивести гроші зі своїх рахунків, банк не зможе виконати свої зобов'язання перед ними. До того ж наявність поняття центрального банку породжує проблему централізації фінансової системи. Така централізованість може призвести до зупинки або затримки роботи усієї грошової системи або її частин, у випадку кризових ситуацій, що стосуватимуться її критичних вузлів. Також такий центральний регулятор може значним чином впливати на фінансовий ринок, визначати розмір мита за правила проведення транзакцій між користувачами, їх мінімальний розмір, відстежувати усі транзакції та збирати фінансові дані користувачів.

У свою чергу розробники Bitcoin запропонували свій варіант вирішення даних проблем. Bitcoin вважається електронною P2P валютою - тобто представленою у вигляді мережі, у якій користувачі можуть надсилати один одному грошові одиниці, без участі третьої сторони. Така валюта має значні переваги над відомими нам фіатними грошима завдяки своїй автономності, анонімності та простоті в обслуговуванні.

Завдяки своїй будові, для того щоб надсилати та отримувати монету, користувачу необхідно мати тільки адресу власного гаманця, та пару ключів - приватний, який знатиме лише власник гаманця, та публічний, той який бачитимуть усі учасники мережі. Для того щоб виконати транзакцію, користувач повинен надіслати у мережу підписане своїм приватним ключем повідомлення, яке містить адресу гаманця отримувача та кількість монет, які він хоче відправити. Такі повідомлення включаються у блок, який потім буде доданий до блокчейну. Блокчейн можна вважати журналом усіх транзакцій, який є підтвердженням легітимності усіх переказів монет між учасниками мережі. Процес формування нового блоку називається майнінгом.

1.1.2 Схема роботи майнінгу криптовалюти Bitcoin

За задумом розробників криптовалюти Bitcoin, усе функціонування мережі забезпечується безпосередньо її користувачами. Таке рішення дозволяє вирішити проблему централізації фінансової системи та необхідності наявності гаранта для її функціонування.

Також, для того щоб монети такої криптовалюти мали цінність, їх кількість має бути обмеженою. За задумами розробників, система реалізована таким чином, щоб всього було випущено не більше 21 мільйона монет. Монети випускаються під час побудови нового блоку в блокчейні. Наведемо більше деталей стосовно того як працює даний процес.

Як згадувалось вище, процес проведення користувачами грошових

транзакцій передбачає публікування повідомлень у мережу Bitcoin. Такі повідомлення формуються у блоки, для кожного з яких обраховується Proof of Work - хеш-послідовність, яка є складною для обчислення, але легкою для перевірки. Система створена таким чином щоб час пошуку хешу для кожного блоку залишався однаковим, а у випадку знаходження правильного хешу одним з майнерів, усі інші учасники мережі могли швидко підтвердити його легітимність. Кожен окремий комп'ютер, який підключено до мережі та який зберігає дані блокчейну, бере участь у перевірці транзакцій і побудові блоків називають нодою. Новий блок додається до блокчейну кожні 10 хвилин. Транзакція вважається легітимною, якщо після блоку в якому вона була записана в блокчейн було додано ще шість блоків.

Інтерес користувачів мережі брати участь у процесі майнінгу зумовлений винагородою, яку вони отримують за проведену роботу. По суті, побудова нового блоку для блокчейну зводиться до вирішення складної криптографічної задачі, що виконується на спеціально програмно-апаратному обладнанні. В процесі побудови нового блоку, кожен майнер вписує до нього ще одну транзакцію, яка не містить відправника, а містить лише отримувача та додає в мережу нові монети. Додавання нових монет є обмеженим та працює за встановленим розробниками механізмом. Додатково, майнери отримують комісію за проведені користувачами транзакції.

1.1.3 Участь ASIC-схем у процесі майнінгу

На самих ранніх етапах запуску мережі Bitcoin майнінг виконувався на допомогою звичайних комп'ютерних процесорів (CPU). Загальна обчислювальна потужність (хешрейт) усіх нодів, що під'єднані до мережі була не дуже великою, а тому такий варіант був можливим. Згодом, коли почала зростати кількість користувачів, під'єднаних мережі, почав зростати і її загальний хешрейт. Для того щоб процес майнінгу залишався

прибутковим, виникла необхідність у застосуванні більш потужних апаратно-програмних рішень. Таким чином почалось використання графічних процесорів GPU, FPGA та ASIC-схем для майнінгу. Додатково, користувачі, що не володіли достатньою кількістю пристроїв, тобто не могли генерувати необхідну кількість хешрейту, щоб процес майнінгу залишався ергономічним, об'єднувались у пули. Пул - певна кількість пристроїв (різних користувачів), з'єднаних між собою для формування однієї ноди, що братиме участь у пошуку нового блоку. У випадку успішного додавання блоку, винагорода за його знаходження розподілялась між учасниками пулу, відповідно до відсоткових значень їх хешрейтів у пулі. Варто зазначити що поширеним також було поняття хмарного майнінгу, коли користувач міг орендувати обчислювальну потужність пристроїв та використовувати її в цілях майнінгу, доєднавшись до пулу.

Саме на цьому етапі знову виникає проблема централізації навколо функціонування мережі Bitcoin. Окремі користувачі мережі почали накопичувати значні обчислювальні потужності, що передбачає собою розміщення та використання спеціальних інтегральних схем (ASIC) разом з усією необхідною для їх роботи інфраструктурою в одному місці. Такі місця називають майнінг фермами, за будовою та принципами роботи вони є схожими на дата-центри. Проблема появи таких майнінг центрів полягає в тому, що ці місця є вразливими з точки зору фізичного втручання, а також для регуляції та оподаткування зі сторони закону.

Додатково слід зазначити, що поширене використання ASIC-схем у мережі Bitcoin зробило майнінг не вигідним для звичайних користувачів. З урахуванням усіх апаратно-технічних пристроїв, які доступні для користування широкому колу користувачів, отримана завдяки ним обчислювальна потужність не покриватиме витрати на електроенергію. Саме тому виникла ситуація, при якій функціонування мережі Bitcoin, підтвердження транзакцій та побудова їх публічного реєстру (блокчейну) залежить від обмеженого кола користувачів.

1.2 Принципи роботи алгоритму Ethash

Одним із найбільш успішних рішень проблеми централізації майнінгу, яка виникла в мережі Bitcoin унаслідок широкого використання ASIC-схем, стала криптовалюта Ethereum. Дана монета запропонувала принципово новий алгоритм доказу виконаної роботи (Proof of Work), головною схемою хешування якого став `Ethash` [10]. Основною метою його створення було зменшення переваг ASIC-схем шляхом перенесення основного навантаження з арифметичних операцій на операції доступу до оперативної пам'яті. Детальний опис алгоритму опубліковано розробниками в офіційній документації Ethereum [2].

Алгоритм `Ethash` побудований на основі попередньої розробки `Dagger-Hashimoto`. У цій схемі було поєднано дві важливі ідеї. Алгоритм `Hashimoto`, що забезпечував так звану I/O-bound поведінку, коли продуктивність майнінгу визначається швидкістю читання великих обсягів даних з пам'яті. Алгоритм `Dagger`, у свою чергу, використовував псевдовипадково згенерований граф даних, доступ до якого був необхідний для виконання майнінгу. Поєднання цих двох підходів дозволило створити алгоритм, стійкість якого до ASIC-оптимізації значною мірою визначалась пропускною здатністю пам'яті, а не кількістю обчислювальних блоків.

1.2.1 Загальна структура алгоритму

Основною ідеєю `Ethash` є використання великого псевдовипадкового набору даних, що має назву DAG (Directed Acyclic Graph). Цей набір даних генерується детерміновано на основі номера блоку та є однаковим для всіх учасників мережі. На відміну від блокчейну Bitcoin, де обчислення зводяться переважно до багаторазового застосування SHA-256, у схемі `Ethash` майнер повинен виконувати численні випадкові

звернення до елементів DAG.

DAG оновлюється кожні 30000 блоків, що відповідає так званій епосі. Для мережі Ethereum, де середній час створення блоку становив приблизно 15 секунд, одна епоха тривала близько 125 годин або 5.2 доби. На початковому етапі роботи мережі розмір DAG становив приблизно 1 ГБ, а надалі поступово збільшувався. Згідно зі специфікацією, початковий розмір датасету дорівнював

$$\texttt{DATASET\_BYTES\_INIT} = 2^{30} \text{ байт},$$

а приріст на кожну епоху становив

$$\texttt{DATASET\_BYTES\_GROWTH} = 2^{23} \text{ байт}.$$

Процес майнінгу починається з формування вхідних даних, що складаються із заголовка блоку та випадкового числа `nonce`. На їх основі за допомогою функції Keccak-512 (історично в документації Ethereum вона позначається як `sha3_512`) обчислюється початкове значення:

$$s = \text{Keccak-512}(H \parallel nonce).$$

Після цього формується структура `mix` розміром 128 байт шляхом дублювання значення s. Далі алгоритм виконує 64 ітерації, на кожній з яких на основі поточного значення `mix` обчислюється новий індекс для звернення до DAG. Вибрані дані комбінуються з поточним станом за допомогою функції FNV:

$$\text{fnv}(v_1, v_2) = ((v_1 \cdot 0x01000193) \oplus v_2) \bmod 2^{32}.$$

Після завершення усіх ітерацій значення `mix` стискається до 32-байтового `mix digest`, а остаточний результат обчислюється за формулою:

$$\text{result} = \text{Keccak-256}(s \parallel mix\_digest).$$

Отримане значення порівнюється з цільовим порогом складності:

$$\text{result} < \text{Target}.$$

У випадку виконання цієї умови знайдений `nonce` вважається правильним, а блок може бути доданий до блокчейну.

1.2.2 Генерація seed, cache та DAG

Для кожної епохи обчислюється 32-байтове seed-значення. На початку роботи алгоритму воно складається з нульових байтів:

$$S_0 = 0^{32}.$$

Для кожної нової епохи до попереднього seed застосовується Keccak-256:

$$S_{i+1} = \text{Keccak-256}(S_i).$$

На основі seed генерується cache — невеликий набір даних, початковий розмір якого становить:

$$\texttt{CACHE\_BYTES\_INIT} = 2^{24} \text{ байт} \approx 16 \text{ МБ}.$$

Cache використовується для побудови DAG. При цьому кожен елемент DAG залежить від 256 псевдовипадкових вибраних елементів cache:

$$\texttt{DATASET\_PARENTS} = 256.$$

Завдяки такій структурі light-клієнти можуть зберігати лише cache і за необхідності відновлювати окремі елементи DAG. Це значно зменшує вимоги до пам'яті при перевірці знайденого Proof of Work. Мережа Ethereum розділяє вузли на декілька видів. Деякі з них називаються light-клієнтами, які не зберігають повну копію блокчейну. Такі клієнти не

виконують обчислення у повному обсязі та існують для швидкої перевірки даних з мінімальним використанням пам'яті, дискового простору та обчислювальних ресурсів.

Важливою особливістю алгоритму `Ethash` є те, що процес перевірки знайденого Proof of Work потребує значно менше ресурсів, ніж сам майнінг. Повний вузол або light-клієнт не зобов'язаний зберігати весь DAG у пам'яті. Для перевірки достатньо мати лише cache, на основі якого можна детерміновано відновити необхідні елементи датасету. Такий підхід дозволив суттєво зменшити вимоги до обладнання вузлів мережі та спростив процес верифікації блоків.

Таке розподілення клієнтів на повні та light-вузли дозволило зробити мережу більш доступною для широкого кола користувачів, зменшивши вимоги до обсягів пам'яті та обчислювальних ресурсів, а відповідно зробило мережу більш децентралізованою. В противагу цьому, головною проблемою, що виникає внаслідок використання ASIC-схем є централізація обчислювальних ресурсів та вузлів мережі.

Додатковою перевагою `Ethash` стало використання змінного набору даних, який оновлюється кожні 30000 блоків. Унаслідок цього спеціалізовані пристрої змушені постійно працювати з новими структурами даних, розмір яких з часом зростає. Це значно ускладнює створення вузькоспеціалізованих схем, оптимізованих під фіксований набір вхідних даних, та збільшує вартість розробки ASIC-пристроїв для даного алгоритму.

1.2.3 Механізм memory-hardness

Ключовою характеристикою `Ethash` є його складність за пам'яттю. У межах одного обчислення алгоритм виконує:

$$\texttt{ACCESSES} = 64$$

звернення до DAG, тобто операцій доступу до пам'яті. Кожне звернення оперує блоком даних розміром:

$$\texttt{MIX\_BYTES} = 128 \text{ байт}.$$

Таким чином, для обчислення одного хешу необхідно зчитати:

$$64 \times 128 = 8192 \text{ байт} \approx 8 \text{ КБ}.$$

Оскільки адреси звернень залежать від результатів попередніх обчислень, передбачити їх наперед неможливо. Це унеможливлює ефективне кешування та робить продуктивність алгоритму безпосередньо залежною від пропускної здатності оперативної пам'яті:

$$\text{Performance} \propto \text{Memory Bandwidth}.$$

На практиці це означає, що навіть наявність великої кількості арифметичних блоків не забезпечує значної переваги, якщо пристрій не здатний швидко виконувати випадкові читання з пам'яті.

1.2.4 Матеріальна база Dagger-Hashimoto

Базою для створення алгоритму `Ethash` став алгоритм `Dagger-Hashimoto`. У початковій версії Ethereum саме він розглядався як основний механізм Proof of Work. Його структура поєднувала концепцію великого псевдовипадкового графа даних (`Dagger`) та I/O-bound (складну за введенням/виведенням даних з пам'яті) схему роботи (`Hashimoto`). Основною метою даного алгоритму було забезпечення трьох властивостей: ASIC-стійкості, можливості перевірки light-клієнтами та складності розподілення роботи.

У подальшому дана схема була спрощена та оптимізована, що привело до створення `Ethash`. При цьому основні ідеї залишились

незмінними: використання великого датасету, псевдовипадкового доступу до його елементів та переважання операцій читання з пам'яті над арифметичними обчисленнями.

1.2.5 Стійкість до ASIC-оптимізації

Алгоритм `Ethash` був спеціально розроблений для зменшення переваг ASIC-схем. Це досягається завдяки трьом основним властивостям. По-перше, для майнінгу необхідний великий обсяг пам'яті, розмір якої постійно збільшується з ростом блокчейну. По-друге, доступ до елементів DAG має псевдовипадковий характер. По-третє, продуктивність визначається швидкістю виконання операцій з пам'яттю, а не обчислювальною потужністю пристрою.

Для прикладу, відеокарта з пропускною здатністю пам'яті 211 ГБ/с має теоретичну межу продуктивності близько 26 MH/s. Реальні показники GPU були близькими до цього значення, що підтверджує memory-bound характер алгоритму.

Незважаючи на це, з часом були створені ASIC-пристрої для `Ethash`. Однак їх перевага над сучасними GPU виявилась значно меншою, ніж у випадку Bitcoin. Це дозволило звичайним користувачам ще тривалий час брати участь у майнінгу за допомогою доступних графічних процесорів.

1.2.6 Перехід Ethereum на Proof of Stake

15 вересня 2022 року в результаті оновлення, відомого як *The Merge*, мережа Ethereum повністю відмовилася від механізму Proof of Work та перейшла на алгоритм Proof of Stake. З цього моменту `Ethash` більше не використовується для захисту мережі Ethereum. Проте сам алгоритм зберігає історичне значення та продовжує використовуватись в інших криптовалютах, що функціонують на основі Proof of Work.

Таким чином, алгоритм `Ethash` став одним із перших практичних прикладів реалізації memory-hard Proof of Work. Його використання продемонструвало, що перенесення основного навантаження з арифметичних операцій на випадкові звернення до пам'яті дозволяє суттєво обмежити переваги спеціалізованих обчислювальних пристроїв та сприяти більш децентралізованому розподілу обчислювальних ресурсів між учасниками мережі.

1.3 Принципи роботи алгоритму Verthash

Алгоритм Verthash є сучасною реалізацією доказу виконаної роботи (Proof-of-Work), розробленою з метою забезпечення пам'яттєвої складності та підвищення стійкості до спеціалізованих обчислювальних пристроїв (ASIC та FPGA). Його концепція базується на використанні великого попередньо згенерованого набору даних, доступ до якого є обов'язковим під час кожної спроби обчислення хешу.

1.3.1 Генерація таблиці даних

Ключовим елементом алгоритму є файл `verthash.dat`, що являє собою таблицю великого розміру (приблизно 1–1.2 ГБ), яка однакова для всіх учасників мережі. Генерація цієї таблиці відбувається детерміновано на основі фіксованого початкового значення (seed), що задається як 16-байтовий масив:

$$S = \texttt{VERTHASHDATSEED}$$

Для кожного індексу i формується вхідний вектор:

$$\mathrm{Input}_i = S \parallel \mathrm{LE32}(i)$$

після чого обчислюється значення:

$$T[i] = \text{SHA3-256}(\text{Input}_i)$$

Отримані значення послідовно записуються у таблицю. Завдяки цьому всі вузли мережі мають ідентичний набір даних, що виключає можливість прихованих оптимізацій та забезпечує узгодженість обчислень.

1.3.2 Майнінг Vertcoin

Процес майнінгу в алгоритмі Verthash [6] базується на інтенсивному використанні цієї таблиці. Нехай H — заголовок блоку розміром 80 байт, який включає значення nonce. На першому етапі формується набір псевдовипадкових значень за допомогою криптографічної хеш-функції SHA3-512:

$$P_k = \text{SHA3-512}(\text{MutateFirstByte}(H, k)), \quad k = 0, \ldots, 7$$

де функція MutateFirstByte змінює перший байт заголовка.

Отримані значення конкатенуються у масив P довжиною 512 байт, який інтерпретується як набір індексів:

$$\text{Index}_j = \text{LE32}(P[4j : 4j + 3]) \bmod N$$

Далі виконується ітеративний процес змішування, в якому використовується акумулятор:

$$A_0 = 0^{32}$$

та на кожній ітерації виконується операція:

$$A_{j+1} = A_j \oplus T[\text{Index}_j]$$

Після завершення всіх ітерацій фінальне значення акумулятора:

$$\text{PoW hash} = A_{final}$$

використовується як результат доказу виконаної роботи. Блок вважається валідним у випадку виконання умови:

$$\text{PoW hash} < \text{Target}$$

Важливою особливістю алгоритму є те, що він передбачає велику кількість випадкових доступів до пам'яті. У процесі майнінгу виконується тисячі операцій читання з таблиці `verthash.dat`, причому індекси цих звернень залежать від значення nonce та заголовка блоку. Це унеможливлює ефективне кешування даних та виключає можливість попереднього обчислення.

Ключовою властивістю алгоритму є його складність за пам'яттю. Для виконання майнінгу необхідний доступ до великого обсягу оперативної пам'яті, що суттєво обмежує ефективність спеціалізованих пристроїв. На відміну від алгоритмів, орієнтованих на обчислювальну потужність, продуктивність Verthash визначається пропускною здатністю пам'яті та швидкістю випадкового доступу до даних.

Додатковим фактором безпеки є використання псевдовипадкових шаблонів доступу до пам'яті. Оскільки індекси залежать від результатів хешування, кожна нова спроба майнінгу генерує нову послідовність звернень до таблиці. Це унеможливлює оптимізацію на рівні апаратного забезпечення та ускладнює створення спеціалізованих пристроїв.

Використання операції XOR для змішування значень гарантує, що фінальний результат залежить від усіх вибраних елементів таблиці. Будь-яка зміна хоча б одного значення призводить до повної зміни результату, що забезпечує криптографічну стійкість алгоритму.

Таким чином, Verthash є прикладом алгоритму доказу виконаної роботи, орієнтованого на складність за пам'яттю, в якому основний

акцент зроблено на використанні великого набору даних та випадкового доступу до нього. Такий підхід дозволяє зменшити переваги спеціалізованих обчислювальних пристроїв та сприяє більш рівномірному розподілу обчислювальних ресурсів між учасниками мережі.

Висновки до розділу 1

У межах даного розділу було наведено основні теоретичні відомості, що дозволяють охарактеризувати проблему ASIC-атак. На прикладі роботи системи Bitcoin показно основні принципи функціонування криптовалют, зокрема висвітлено поняття майнінгу, алгоритму консенсусу Proof of Work, участь ASIC-схем у процесах генерації нових монет та валідації транзакцій. Наведено приклади ASIC-стійких схем хешування типу Dagger-Hashimoto (Ethash та Verthash).

Отримана в даному розділі інформація дасть змогу перейти до узагальнення знань стосовно ASIC-стійких схем хешування та побудови математичної моделі алгоритму Verthash.

2 ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ АЛГОРИТМУ ХЕШУВАННЯ VERTHASH

У межах даного розділу висвітлено процес побудови математичної моделі алгоритму хешування Verthash. Для побудови моделі був використаний програмний код з реалізацією алгоритму безпосередньо його розробниками, теоретичні викладення, опубліковані ними та публікацію статті за тематикою [11].

2.1 Підходи до побудови моделей хеш-функцій

Як було вказано розробниками, функція хешування Verthash була побудована на засадах та основних принципах роботи схеми хеш-функції Ethash. Хеш-функція криптовалюти Ethereum була створена для її алгоритму консенсусу Proof-of-Work, ціллю якого мав би стати принципово новий підхід до вирішення наявної проблеми використання ASIC-схем. На момент створення алгоритму Ethash, було відомо декілька ключових ідей у конструюванні алгоритмів консенсусу Proof of Work, зокрема ідеї, запропоновані в межах алгоритмів Dagger та Hashimoto. Дані алгоритми містили певні недоліки, якщо розглядати їх роботу окремо.

Алгоритм Hashimoto містить більше практичної цінності, якщо розглядати його як генератор вказівників для роботи з великим об'ємом даних, а алгоритм Dagger є вразливим для ASIC-схем, що містять доступ до загального пулу пам'яті. Алгоритмом, який об'єднав найкращі сторони Dagger та Hashimoto став алгоритм Dagger-Hashimoto. Згодом даний алгоритм було покращено і випущено під назвою Ethash, як основу Proof-of-Work криптовалюти Ethereum.

Hashimoto I/O-bound. Алгоритм Hashimoto був обраний як одна

зі складових частин алгоритму Dagger-Hashimoto, оскільки він був орієнтований на уникнення ASIC-вразливості шляхом перетворення операцій читання з пам'яті, а не обчислювальної потужності. Даний алгоритм використовує блокчейн як джерело даних, до якого необхідний доступ під час роботи.

Головною мотивацією створення даного алгоритму була проблема домінування ASIC-схем у мережі Bitcoin, що призвело до того, що звичайний користувач фактично втратив можливість брати участь у процесі майнінгу. Це суттєво підірвало децентралізований характер криптовалютних мереж.

З технічної точки зору, Hashimoto не є класичним алгоритмом Proof of Work, де винагорода отримується за виконані обчислення. Автор визначав його як алгоритм, обмежений пропускною здатністю введення/виведення даних, тобто I/O-bound. Це означає, що майнінг на основі Hashimoto спирається на процес псевдовипадкового вибору елементів з великого спільного набору даних. Такий підхід унеможливлює перерозподілення обчислень для вузлів, що не мають доступу до повного набору даних, та може використовуватись не як самостійний алгоритм Proof of Work, а як генератор вказівників на дані у блокчейні. Таким чином, усі вузли, що беруть участь у процесі підтвердження блоків, змушені самостійно верифікувати транзакції.

Проблема перерозподілення задач підтвердження блоків безпосередньо пов'язана з функціонуванням майнінг-пулів, що були описані у попередньому розділі. Учасники пулу фактично позбавлені можливості знати, які саме транзакції підтверджуються їхніми обчислювальними потужностями. Такий підхід завдає шкоди децентралізації мережі та перетворює майнінг виключно на спосіб отримання прибутку, що конфліктує з самою суттю блокчейну як надійного та децентралізованого реєстру.

Dagger. Алгоритм Dagger був розроблений як альтернатива Scrypt, що забезпечує складні за витратами пам'яті обчислення та швидку

верифікацію. Подібно до Hashimoto, Dagger орієнтований на стійкість проти ASIC-атак. Стійкість цього алгоритму зумовлена вимогами володіти значними обсягами оперативної пам'яті для обчислень. Така кількість обчислювальних ресурсів необхідна для роботи з значним обсягом даних, реалізованим у вигляді направлених ациклічних графів (DAG), від абревіатури яких і названо сам алгоритм. DAG це граф, що не містить направлених циклів між своїми вершинами, тобто починаючи обхід з будь-якої вершини, неможливо повернутись до неї знову. Згідно з whitepaper документом алгоритму, для його виконання потрібно 512 МБ пам'яті, тоді як валідація потребує лише 112 КБ та 4078 хешів. Саме тому визначальним фактором складності майнінгу є саме пам'ять, а не обчислювальна потужність.

Завдяки описаним вище властивостям, спеціалізоване апаратне забезпечення мало досить незначну перевагу при майнінгу в мережах, з алгоритмами на основі Dagger. Проте вразливість алгоритму виразилась у можливості паралелізації обчислень за умови підключень декількох обчислювальних пристроїв до спільного пулу пам'яті. Принцип роботи Proof of Work у алгоритмі Dagger базується на генерації псевдовипадкового набору даних з ціллю заповнення оперативної пам'яті майнера, що відбувається у декілька раундів. На початку кожного нового раунду певна кількість елементів, отриманих на виході попереднього раунду, хешується разом. Такий підхід дозволив виробникам ASIC-схем оптимізувати їх роботу в мережах Dagger шляхом організації спільного доступу до пам'яті між пристроями та збереження результатів кожного раунду, що суттєво спрощувало генерацію нових наборів даних для ASIC-схем порівняно зі звичайними комп'ютерами. Саме ця вразливість стала причиною відмови від Dagger як самостійної хеш-функції.

Dagger-Hashimoto реалізує підхід Hashimoto для забезпечення складності введення/виведення даних, проте, на відміну від оригінального Hashimoto, не використовує блокчейн як набір даних. Замість цього, алгоритм генерує власний датасет розміром 1+ ГБ,

побудований за принципом алгоритму Dagger. Вразливість Dagger до атак через встановлення спільного доступу до пам'яті була усунена шляхом використання змінного, не постійного набору даних (датасету), який оновлюється через заздалегідь визначені проміжки часу. Таке рішення робило витрати на генерацію нового набору даних незначними та вирішило попередню проблему оптимізації ASIC-схем.

2.2 Генерація таблиці даних

Алгоритм Verthash став покращенням попередньо відомого рішення проблеми ASIC-атак - алгоритму Ethash. Розробники монети Ethereum вирішили змінити внутрішню політику та відійшли від принципу забезпечення ASIC-стійкості, перейшовши на інший алгоритм консенсусу Proof of Stake. На противагу цьому, розробниками монети Vertcoin було прийнято рішення продовжувати боротьбу з оновлюваними ASIC-схемами, шляхом постійного покращення функції хешування, що лежить в основі протоколу Proof of Work. Алгоритм Verthash став удосконаленішою версією алгоритму Ethash, покращуючи його роботу в тих місцях, де це можливо. Одним з них є процес генерації таблиці даних, необхідних для роботи алгоритму. На подальших кроках роботи алгоритму, ця таблиця бере участь у процесі майнінгу.

Безпосередньо самі розробники називають алгоритм Verthash «датасет-важким», оскільки він вимагає постійної роботи з таблицею даних, що в свою чергу можливе завдяки використанню значних обсягів оперативної пам'яті для рандомізованого доступу до частин цієї таблиці. Така схема забезпечує стійкість системи до ASIC-атакою Схема, зображена на рисунку 2.1 демонструє процес генерації таблиці даних, який відбувається перед кожним раундом блокчейну.

Початковим компонентом у даній схемі є значення seed, яке є ASCII-представленням текстового рядка «VERTHASHDATSEED», доповнене нульовим байт-термінатором вкінці. Наступною компонентою є

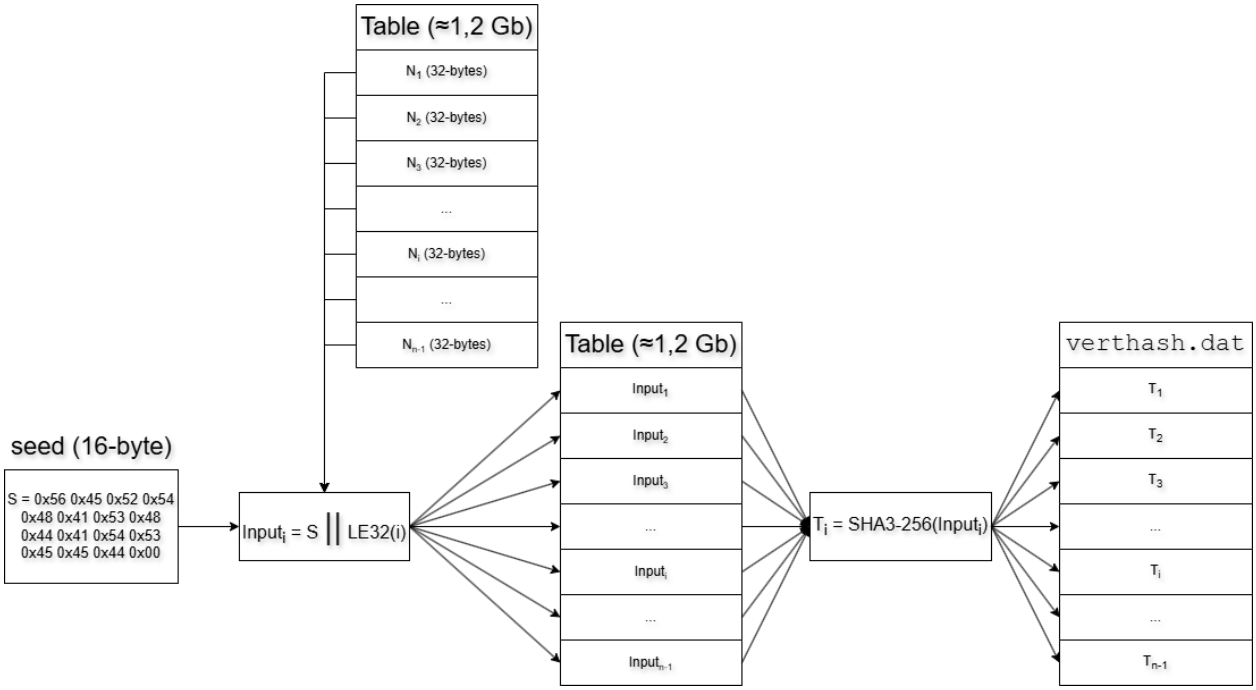

Рисунок 2.1 – Генерація табличних даних

таблиця даних, у якій міститься N входжень. Значення seed по черзі конкатенується з індексовим значенням N-го входження таблиці, записаного у 4-байтовому little-endian форматі. Далі, до усіх отриманих в минулому кроці табличні даних застосовується алгоритм хешування SHA3-256, і кожне з входження знову записується в таблицю. Цей масив даних і є наповненням файлу verthash.dat, який є однаковим для усіх користувачів блокчейну. Сам процес майнінгу є безпосередньо залежним від цього файлу, оскільки необхідним є рандомізований доступ до його частин verthash.dat під час побудових нових блоків блокчейну.

2.3 Схема майнінгу за допомогою алгоритму Verthash

Схема майнінгу криптовалюти vertcoin 2.2 комбінує в собі як підходи до побудови хеш-функцій за Меркле-Дамгардом [5, 3], так і схеми хешування типу Sponge [5, 3]. Хешування за допомогою алгоритму verthash відбувається у декілька етапів. Спочатку генерується набір індексів, за допомогою алгоритму хешування SHA3-512, який є

представником схем хешування типу Sponge. Для побудови нового блоку дана схема приймає на вхід заголовок (Header) з попереднього блоку (послідовність довжиною 80 байтів), а на вхід подає масив індексів, за допомогою якого організовується доступ до частин таблиці `verthash.dat` Процес створення набору індексів є рандомізованим, а тому не може бути обчисленим наперед. Тобто процедура доступу до табличних даних, що зберігається у пам'яті забезпечує складність роботи схеми відносно операцій з пам'яттю (memory hard).

Завершальним етапом роботи схеми 2.2 побудови нового блоку монети vertcoin є міксинг. Даний процес є ітеративним (128 ітерацій), у ньому кожне наступне входження залежить від обрахунку попереднього значення, що безпосередньо відповідає принципам побудови хеш-функцій за схемою Меркле-Дамгарда.

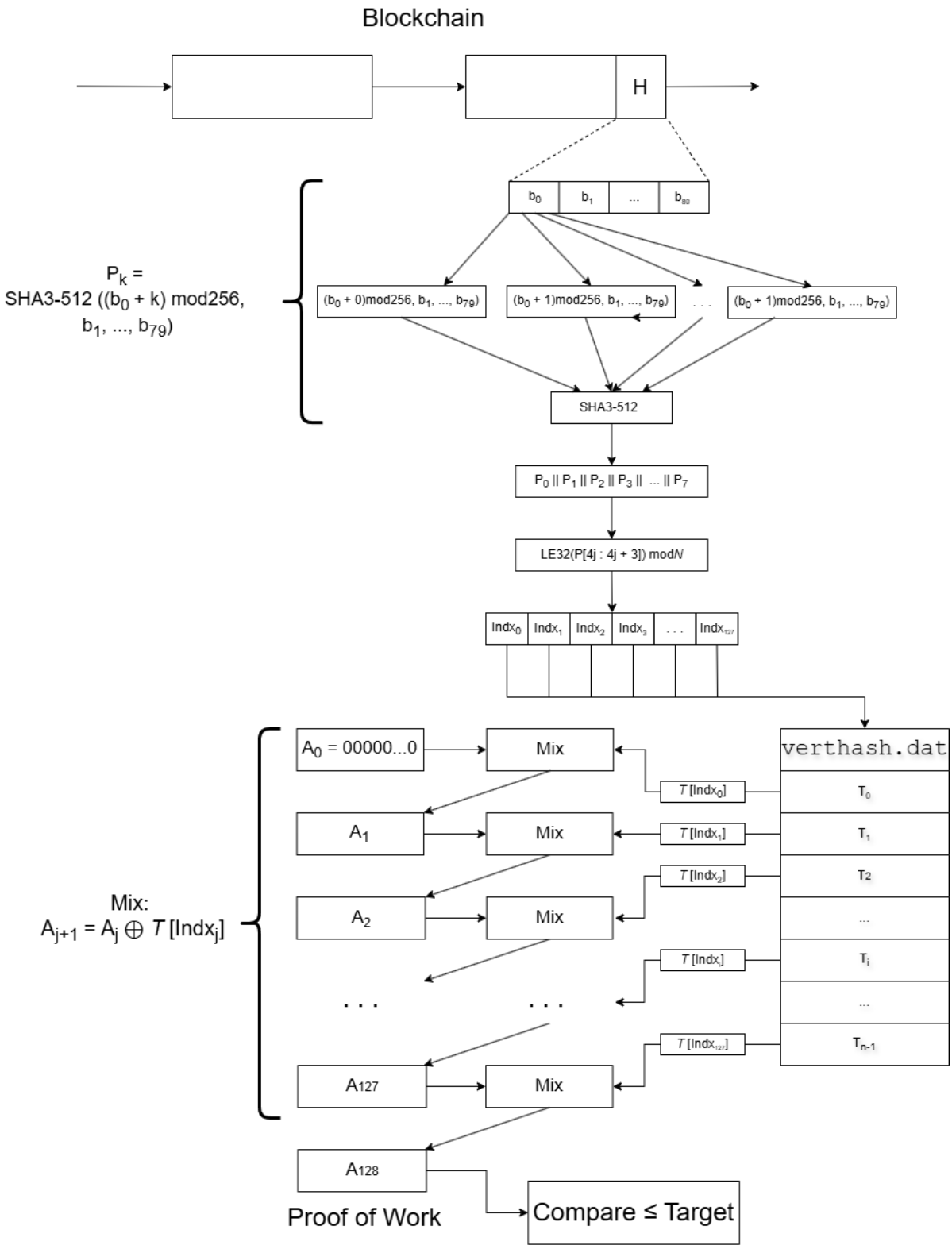

Рисунок 2.2 – Схема майнінгу блоку транзакцій Vertcoin

У наведеній схемі 2.2 відображено спрощену математичну структурну схему алгоритму Verthash, у якій процес змішування виконується протягом

128 ітерацій. Реальна програмна реалізація алгоритму додатково розширює масив вказівників та виконує 4096 звернень до файлу verthash.dat.

2.4 Побудова структурної схеми хешування Verthash

Схематично модель роботи алгоритму хешування Verthash може бути представлена у вигляді двох основних етапів. Перший етап передбачає формування набору індексів, які визначають порядок звернення до великого набору даних. Для цього використовується заголовок блоку, який обробляється функцією хешування. Отримані результати інтерпретуються як послідовність числових значень, що надалі використовуються як адреси доступу до елементів таблиці.

Другий етап полягає у послідовному зчитуванні даних за сформованими індексами та виконанні операцій міксування. На кожній ітерації поточне значення акумулятора комбінується зі значенням, отриманим із таблиці. Результат кожної операції використовується як вхідне значення для наступної ітерації. Таким чином формується послідовність взаємопов'язаних обчислень, у якій кожен наступний крок залежить від результатів попереднього. У результаті роботи алгоритму формується підсумкове значення, яке порівнюється з цільовим параметром `target` для доказу виконаної роботи Proof of Work (рис. 2.3).

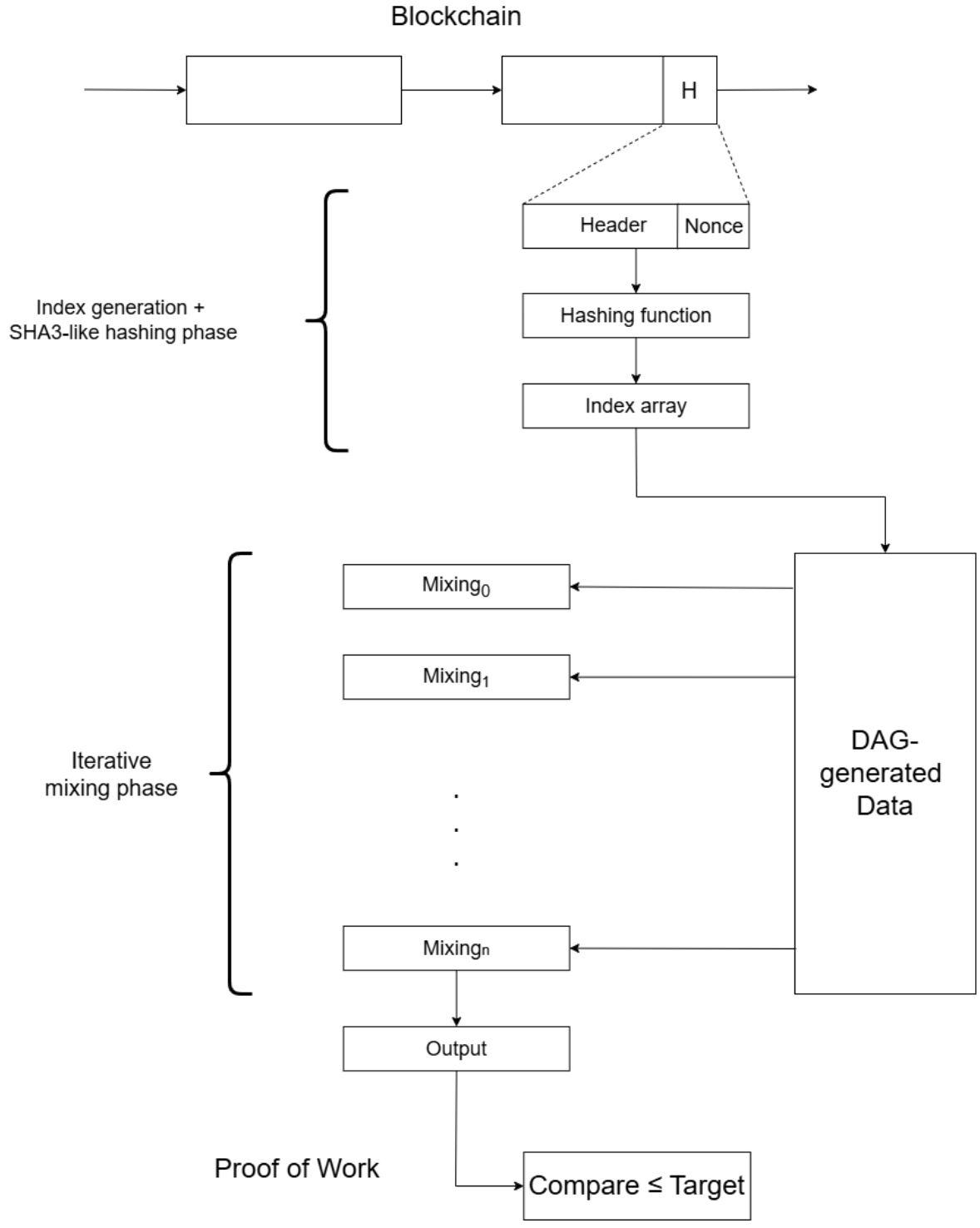

Рисунок 2.3 – Модель структурної схеми роботи алгоритму хешування Verthash

Варто зазначити, що наведена структурна схема не є єдиним можливим способом реалізації ASIC-стійкої схеми хешування. Існують алгоритми, у яких формування індексів та процес міксування не

розділяються на окремі етапи. У такому випадку адреси доступу до елементів великого набору даних обчислюються безпосередньо під час виконання циклу міксування.

Подібний підхід призводить до того, що порядок звернення до пам'яті визначається результатами попередніх обчислень. Це унеможливлює попереднє передбачення адрес доступу до даних та значно ускладнює ефективне кешування. Унаслідок цього продуктивність алгоритму стає безпосередньо залежною від пропускної здатності пам'яті, а не лише від швидкості виконання арифметичних операцій. Схематичне представлення такого підходу наведено на рис. 2.4.

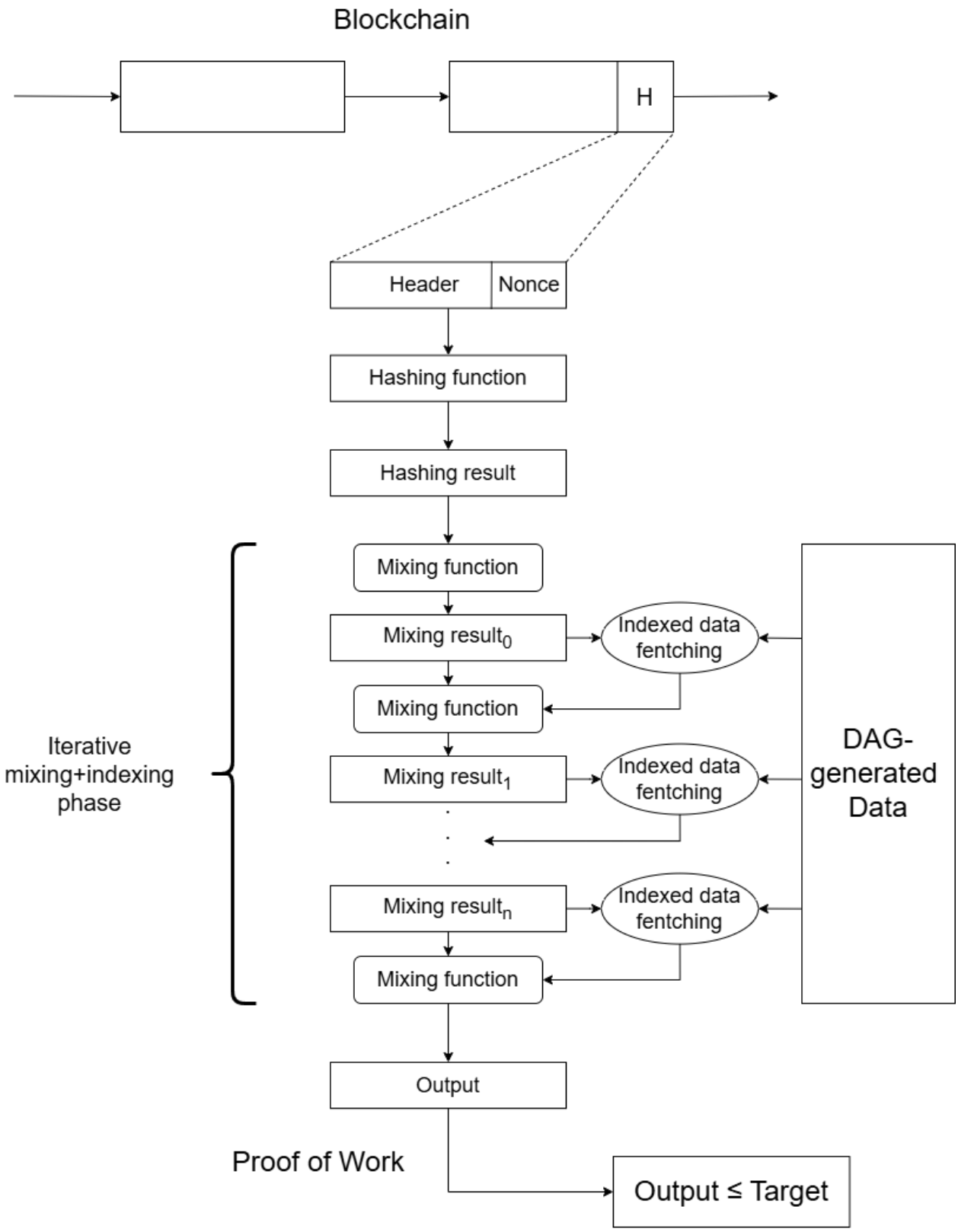

Рисунок 2.4 – Модель альтернативної структурної схеми роботи алгоритму хешування

2.5 Порівняння алгоритмів Hashimoto, Ethash та Verthash

Одним з перших варіантів розв'язання проблеми застосування ASIC-схем для майнінгу був алгоритм Hashimoto [7]. У схемі даного алгоритму вперше була реалізована ідея використання датасету утвореного з складових блокчейну в якості великого набору даних для своєї роботи. Обмежувальним фактором, що запобігає проведенню ASIC-атак є процес зчитування датасету з пам'яті.

Процес роботи даного алгоритму потребує повного доступу до датасету, оскільки схема роботи Hashimoto передбачає псевдовипадкове зчитування його частин. Це означає що розподілення обсягу роботи даного алгоритму між нодами буде мало ефективним. В такому випадку кожній з них доведеться локально тримати в пам'яті даний датасет.

У схемах роботи даного алгоритму можна виділити два основних етапи: етап створення масиву вказівників для роботи з датасетом та етап міксингу. Саме у внутрішніх процесах даних складових алгоритму і криється відмінність їх роботи. Наприклад, алгоритм ethash розширює підхід Hashimoto до створення вказівників, використовуючи SHA3-подібну функцію хешування та складніші механізми доступу до пам'яті. Використання таких підходів допомагає підвищити загальну ASIC-стійкість схеми.

Алгоритм verthash став оновленою версією алгоритму ethash, покращивши ASIC-стійкість його схеми. Зокрема, ethash використовує 64 операції читання з пам'яті, у межах яких обробляються блоки даних розміром 128 байтів, тоді як Verthash використовує 128 ітерацій із блоками по 256 байтів. Тобто ethash використовує 8 Кб пам'яті для обчислення одного хешу, а verthash - 32 Кб (Таблиця 2.1).

Таблиця 2.1 – Порівняльний аналіз технічних та криптографічних характеристик пам'яттєво-залежних алгоритмів хешування

Критерій порівняння	**Dagger-Hashimoto**	**Ethash**	**Verthash**
Базова хеш-функція	Keccak-512 та SHA-256	Keccak-256 та Keccak-512	SHA3-256
Структура даних пам'яті	Направлений ациклічний граф (DAG) + Dataset	Великий граф (DAG-файл), що циклічно оновлюється	Статична таблиця псевдовипадкових даних (`verthash.dat`)
Об'єм даних для пам'яті	Динамічний (початково в межах сотень МБ)	Динамічний, зростаючий з кожною епохою (від 1 до $>$ 6 ГБ)	Фіксований розмір $\approx$ 1.15 ГБ (не залежить від висоти блока)
Періодичність оновлення	Кожні 30 000 блоків (нова епоха)	Кожні 30 000 блоків (перегенерація DAG)	Не оновлюється (генерується один раз при ініціалізації)
Функція змішування	Модульне підсумовування та бітові зсуви	FNV-1 та логічні операції XOR	Модифікована функція FNV-1a (32-біт) та XOR
Звернення до пам'яті	64 ітерації на один хеш	64 ітерації (зчитування блоками по 128 байт)	40 ітерацій псевдовипадкового блукання по масиву
Головний фактор ASIC-стійкості	Високі вимоги до об'єму та швидкості кеш-пам'яті	Обмеження пропускною здатністю шини пам'яті (Memory Bandwidth Bound)	Жорстке обмеження пропускної здатності пам'яті та економічна недоцільність розробки чипів

Висновки до розділу 2

У даному розділі побудовано структурну схему ASIC-стійкого алгоритму хешування Verthash. Під час виконання даного завдання було підібрано необхідний математичний апарат та відповідні базові математичні моделі хеш функцій, необхідні для побудови даної структурної схеми. Виконано порівняльний аналіз I/O-bound стійких алгоритмів хешування (Hashimoto, Ethash, Vertcoin). Результати подано у вигляді таблиці.

3 АНАЛІЗ ВЛАСТИВОСТЕЙ ASIC-СТІЙКОСТІ АЛГОРИТМУ VERTHASH

Майнінг монети vertcoin продовжує функціонувати, зберігаючи стійкість до ASIC-атак. У межах даного розділу розглянуто основні безпекові властивості забезпечення ASIC-стійкості алгоритмом Verthash, висвітлено підходи до подолання проблеми інтеграції ASIC-схем у обчислювальні потужності мереж криптовалют.

3.1 Зміна функцій хешування мережі Vertcoin

Розробниками монети Vertcoin було виконано декілька поступових змін головної функції хешування алгоритму консенсусу Proof of Work. Інформація про зміни в роботі мережі (хардфорки) криптовалюти наведено в таблиці 3.1. Така необхідність була викликана змаганнями між розробниками ASIC-схем та розробниками хеш-функцій для алгоритмів консенсусу Proof of Work. Створення нових конфігурацій ASIC-схем, що дають змогу прискорити обчислення нового блоку у мережах блокчейн, призводять до створення нових алгоритмів, які є складнішими для паралелізації та апаратного прискорення, і навпаки.

Таблиця 3.1 – Зміна алгоритмів Proof of Work у Vertcoin

Назва алгоритму	**Рік переходу мережі на новий алгоритм хешування**	**Коротка характеристика**
`Scrypt-N`	2014	Модифікація Scrypt з залежністю за часом, вимоги до обсягу пам'яті та обчислювальних ресурсів підвищуються поступово.
`Lyra2RE`	2014	Почергове використання кількох хеш-функцій, завершуючи хеш-функцією Lyra2, типу sponge.
`Lyra2REv2`	2015	Оновлена версія Lyra2RE з модифікованим набором хеш-функцій, посилення орієнтації на GPU-майнінг.
`Lyra2REv3`	2019	Модифікація функції Lyra2, що збільшує використання пам'яті та ускладнює реалізацію на FPGA та ASIC.
`Verthash`	2021	Складний за операціям введення/виведення з пам'яті алгоритм, який псевдовипадковий доступ до елементів датасету (файлу `verthash.dat`).

На відміну від розробників монети Ethereum, розробники монети Vertcoin обрали стратегію розвиту в напрямку протидії ASIC-атакам [8]. Перехід на алгоритм консенсусу Proof of Stake, обраний в якості альтернативи Proof of Work для монети Ethereum не вирішив проблему централізації.

Даний алгоритм консенсусу влаштований таким чином, що гравець, який має найбільшу кількість монет, має пріоритет бути обраним у якості гаманця для проведення транзакцій монет. Такий підхід має ризики

появи на ринку «великого» гаманця, який проводитиме більшість транзакцій, підриваючи децентралізований характер мережі. У такому випадку контроль над підтвердженням блоків може зосередитися в руках обмеженої кількості учасників, які володіють найбільшими запасами криптовалюти.

Крім того, алгоритм Proof of Stake не потребує виконання складних обчислень, а отже не використовує криптографічні хеш-функції як основний механізм забезпечення безпеки мережі. Замість цього безпека системи ґрунтується на економічній зацікавленості власників монет у підтриманні коректної роботи мережі. Такий підхід є ефективним з точки зору енергоспоживання, проте не побудований на використанні криптографічних властивостей хеш-функцій для забезпечення безпеки мереж.

3.2 Забезпечення ASIC-стійкості в мережі Vertcoin

Однією з головних цілей створення криптовалюти Vertcoin було забезпечення можливості участі у майнінгу для широкого кола користувачів без необхідності придбання спеціалізованого обладнання. На відміну від мережі Bitcoin, де процес майнінгу з часом став практично повністю залежним від використання ASIC-схем, розробники Vertcoin послідовно впроваджували алгоритми, орієнтовані на використання звичайних графічних процесорів. У результаті цього було сформовано підхід, у якому ASIC-стійкість мережі забезпечується не одним окремим технічним рішенням, а поєднанням декількох взаємопов'язаних механізмів.

Архітектурна залежність від сучасних комп'ютерних систем. Основним механізмом забезпечення ASIC-стійкості у сучасній реалізації Vertcoin є використання алгоритму Verthash. Принцип його роботи базується на використанні великого набору даних, доступ до якого здійснюється у псевдовипадковому порядку. У процесі майнінгу

виконується значна кількість звернень до елементів файлу `verthash.dat`, а продуктивність алгоритму визначається насамперед швидкістю роботи підсистеми пам'яті.

Такий підхід безпосередньо спирається на архітектуру сучасних комп'ютерних систем. Графічні процесори вже оснащені високошвидкісною відеопам'яттю та широкими шинами передачі даних, що дозволяє їм ефективно виконувати memory-hard алгоритми. Для створення ASIC-пристрою, який забезпечував би суттєву перевагу, необхідно фактично відтворити аналогічну підсистему пам'яті, включаючи значний її обсяг, високу пропускну здатність та механізми випадкового доступу до даних.

У такому випадку перевага спеціалізованого пристрою істотно зменшується, оскільки основна частина його вартості припадає не на арифметичні блоки, а на реалізацію пам'яті. Фактично розробник ASIC змушений будувати систему, архітектурно близьку до сучасного GPU. Таким чином, для досягнення значного приросту продуктивності необхідно не лише оптимізувати окремі логічні елементи, а й створити принципово іншу архітектуру обчислювальної системи, що є складним та економічно невигідним завданням.

Гнучкість розробників у зміні правил роботи мережі. Іншим важливим механізмом забезпечення ASIC-стійкості є можливість оперативної зміни алгоритму майнінгу шляхом проведення хардфорку. Протягом своєї історії Vertcoin вже декілька разів змінював алгоритм Proof of Work, переходячи від Scrypt-N до Lyra2RE, Lyra2REv2, Lyra2REv3 та, зрештою, Verthash.

Кожен такий перехід був відповіддю на появу нових способів апаратної оптимізації або на виявлення потенційних загроз централізації майнінгу. Якщо для певного алгоритму буде створено ASIC-пристрій, розробники можуть змінити правила роботи мережі та впровадити нову схему хешування. Після цього раніше розроблене обладнання втратить свою практичну цінність, оскільки більше не відповідатиме вимогам

оновленого протоколу.

Наявність такого механізму суттєво знижує мотивацію інвестувати значні кошти у створення спеціалізованих пристроїв. Навіть у випадку успішної розробки ASIC не існує гарантії, що дане обладнання зможе використовуватись протягом тривалого часу. Таким чином, хардфорк виступає не лише інструментом оновлення мережі, а й важливим економічним фактором стримування централізації майнінгу.

Висока вартість розробки спеціалізованих пристроїв. Розробка сучасної ASIC-схеми потребує значних фінансових витрат, пов'язаних із проектуванням архітектури, моделюванням, верифікацією, виготовленням дослідних зразків та запуском серійного виробництва. Загальна вартість таких робіт може сягати мільйонів доларів США.

У випадку Verthash економічна доцільність таких інвестицій додатково зменшується через обмежений розмір ринку. Vertcoin не належить до криптовалют з найбільшою капіталізацією, а тому потенційний прибуток від використання спеціалізованого обладнання є суттєво нижчим, ніж у випадку найбільших мереж. Крім того, відсутня гарантія, що алгоритм залишиться незмінним протягом часу, достатнього для окупності вкладених коштів.

Поєднання високої вартості розробки, обмеженого економічного потенціалу та ризику втрати актуальності обладнання створює суттєвий бар'єр для виробників ASIC. У багатьох випадках витрати на проектування та виготовлення такого пристрою перевищують очікуваний прибуток від його експлуатації.

Таким чином, ASIC-стійкість мережі Vertcoin забезпечується одночасним використанням технічних та економічних механізмів. Алгоритм Verthash переносить основне навантаження на підсистему пам'яті та використовує архітектурні особливості сучасних комп'ютерних систем. Можливість проведення хардфорків дозволяє оперативно змінювати правила роботи мережі у відповідь на появу спеціалізованого обладнання. Висока вартість розробки ASIC та невизначеність щодо

строків його окупності додатково знижують зацікавленість у створенні таких пристроїв. Сукупність зазначених факторів дозволяє підтримувати більш справедливий та децентралізований розподіл обчислювальних ресурсів між учасниками мережі.

Висновки до розділу 3

У межах даного розділу розглянуто особливості забезпечення ASIC-стійкості схеми хешування Verthash. Проведено порівняльний аналіз підходів до протидії ASIC-атакам на прикладі алгоритмів консенсусу криптовалют Vertcoin та Ethereum. Висвітлено основні аспекти забезпечення ASIC-стійкості з практичної точки зору.

ВИСНОВКИ

Для виконання основних завдань даної наукової роботи було проведено дослідження роботи схеми хешування Verthash. Основними складовими, які стали предметною базою для дослідження стали складні за введенням/виведенням з пам'яті функції хешування типу Dagger-Hashimoto.

Серед виконаних завдань даного наукового дослідження є проведення детального аналізу опублікованих джерел за тематикою роботи, а саме принципів побудови та функціонування алгоритмів консенсусу мереж криптовалют, застосування ASIC-схем для майнінгу криптовалют та засоби протидії їм. Отримана теоретична база дала змогу перейти до виконання подальших завдань наукової роботи, зокрема побудови моделі ASIC-стійкої схеми хешування Verthash та аналізу її криптографічних властивостей.

В процесі виконання завдань даного наукового дослідження було обрано відповідний математичний апарат для правильної побудови моделі ASIC-стійкої схеми хешування Verthash та систематизації знань стосовної функціонування схем даного типу.

Результати виконання завдання побудови моделі алгоритму Verthash представлені у вигляді графічних схем з переліком теоретичних вимог та принципів стосовно побудови її схеми. Усі отримані схеми побудовані з урахуванням базових математичних моделей хешфункцій типу sponge (при генерації базового масиву даних) та ітеративних схем хешування.

Побудовані в роботі схеми дають змогу більш формально досліджувати криптографічні властивості ASIC-стійких схем хешування, що і дає базу для проведення подальших досліджень за даною тематикою. Результати даного дослідження можуть задати напрямок для побудови нових схем, направлених на унеможливлення застосування

ASIC-схем у майнінгу криптовалют.

ПЕРЕЛІК ПОСИЛАНЬ

[1] *Bitcoin Wiki*. Англ. Bitcoin Wiki. 2026. URL: https://en.bitcoin.it/wiki/Main_Page (дата зверн. 05.02.2026).

[2] Vitalik Buterin. *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform*. Англ. 2014. 36 с. URL: https://ethereum.org/whitepaper/.

[3] Jean-Sébastien Coron та ін. «Merkle-Damgård Revisited: How to Construct a Hash Function». Англ. В: *Advances in Cryptology – CRYPTO 2005*. За ред. Victor Shoup. Т. 3621. Lecture Notes in Computer Science. Springer, 2005. URL: https://iacr.org/archive/crypto2005/36210424/36210424.pdf.

[4] Wei Dai. *b-money*. Англ. 1998. URL: http://www.weidai.com/bmoney.txt.

[5] Boneh Dan та Shoup Victor. *A Graduate Course in Applied Cryptography*. Англ. 2017. 818 с. URL: https://crypto.stanford.edu/~dabo/cryptobook/BonehShoup_0_4.pdf.

[6] Vertcoin Developers. *Vertcoin: Algorithmic Adaptation and the Pursuit of Equitable Proof-of-Work*. Англ. 2025. URL: https://vertcoinproject.org/vertcoin_whitepaper.pdf.

[7] Thaddeus Dryja. *Hashimoto: I/O bound proof of work*. Англ. 2014. 5 с. URL: https://diyhpl.us/~bryan/papers2/bitcoin/meh/hashimoto.pdf.

[8] jk_14r. *Discussion about Verthash ASIC resistance*. Англ. Reddit. 2018. URL: https://www.reddit.com/r/vertcoin/comments/cbviq2/comment/etl0w48/.

[9] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Англ. SSRN Electronic Journal, 2008. 9 с. URL: https://ssrn.com/abstract=3440802.

[10] Vijay Pradeep. *Ethereum's Memory Hardness Explained*. Англ. VijAY PRADEEP Blog. 2017. URL: https://www.vijaypradeep.com/blog/2017-04-28-ethereums-memory-hardness-explained (дата зверн. 12.03.2026).

[11] Дмитро В. Молдован та Наталія В. Кучинська. «Побудова та узагальнення моделі стійкої до ASIC-атак хеш-функції Verthash. Матеріали XXIV Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених». В: *Теоретичні і прикладні проблеми фізики, математики та інформатики*. Навчально-науковий фізико-технічний інститут КПІ ім. Ігоря Сікорського. Київ, Україна, 2026, с. 414—417. URL: https://drive.google.com/file/d/1DYPlUIIpPA9vB1UJqO42TBHmg3jaJ6ZW/view (дата зверн. 19.05.2026).