Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
завідувачка кафедри кібербезпеки
та захисту інформації
_____Наталія ЛУКОВА-ЧУЙКО
«14» червня 2022р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

дипломної роботи
бакалавра
(назва освітнього ступеня)

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)

спеціальність 125 Кібербезпека
(код і назва спеціальності)

освітня програма Кібербезпека
(назва освітньої програми)

на тему: «Дослідження технології блокчейн в криптовалютній індустрії»

Виконавець: студент IV курсу, групи КБ-41

_____ **Аліна КЛЕЩЕНКО**
(підпис) (прізвище ім'я по-батькові)

	Прізвище, ініціали	**Підпис**
Керівник	Андрій ФЕСЕНКО	

Нормоконтроль	Олександр ТОРОШАНКО	

Київ 2022

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
завідувачка кафедри кібербезпеки
та захисту інформації
_____Наталія ЛУКОВА-ЧУЙКО
«01» листопада 2021 р.

ЗАВДАННЯ
на виконання дипломної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньої програми)

Студентові **КБ-41** **Клещенко Аліна Олексіївна**
(група) (прізвище ім'я по-батькові)

Тема дипломної роботи Дослідження технології блокчейн в криптовалютній індустрії

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема дипломної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №5 від 29.10.2021 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Структура, види і принципи роботи технології блкчейн, типи консенсусу, класифікація криптовалют і стеблкоїнів

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Необхідно ознайомитися з теорією технології блокчейн, її принципами роботи, розглянути поняття і види консенсусу, зробити аналіз блокчейн-платформ, їх переваги, недоліки і практична цінність, дослідити особливості криптовалют і порівняти їх між собою, розробити програмну реалізацію технології блокчейн

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Реалізація технології блокчейн в криптовалюті та визначення сфер застосування

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 01 листопада 2021 року

Завдання видав _____ (підпис) Андрій ФЕСЕНКО (ініціали, прізвище)

Завдання прийняв до виконання _____ (підпис) Аліна КЛЕЩЕНКО (ініціали, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	01.11.2021 – 30.01.2022	*виконано*
2	Аналіз літератури	31.01.2022 – 21.02.2022	*виконано*
3	Розгляд історії виникнення і розвитку технології блокчейн	22.02.2022 – 1.03.2022	*виконано*
4	Дослідження принципи роботи технології блокчейн	2.03.2022 – 22.03.2022	*виконано*
5	Аналіз видів консенсусу	23.03.2022 – 05.04.2022	*виконано*
6	Дослідження блокчейн-платформ	06.04.2022 – 14.04.2022	*виконано*
7	Розгляд особливостей криптовалют	15.04.2022 – 04.05.2022	*виконано*
8	Дослідження стейблкоїнів, їх класифікація і приклади	05.05.2022 – 17.05.2022	*виконано*
9	Програмна реалізація блокчейн	18.05.2022 – 02.06.2022	*виконано*
10	Оформлення пояснювальної записки	03.06.2022 – 05.06.2022	*виконано*
11	Підготовка до захисту	06.06.2022 – 13.06.2022	*виконано*

Завдання видав _____ (підпис) Андрій ФЕСЕНКО (ініціали, прізвище)

Завдання прийняв до виконання _____ (підпис) Аліна КЛЕЩЕНКО (ініціали, прізвище)

Термін подання дипломної роботи до ЕК 06 червня 2022 року

РЕФЕРАТ

Дипломна робота складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел та додатку. Основний текст займає 93 сторінки, включає в себе: зміст, вступ, три розділи дипломної роботи, висновки, список джерел та 2 додатки із кількістю сторінок 24. У дипломній роботі міститься 22 рисунків, 1 таблиця, список використаних джерел містить 87 найменування і займає 8 сторінок.

Метою дипломної роботи є реалізація блокчейна на основі національного стандарту ДСТУ 7564:2014, щоб використати його для благодійності на державному рівні.

Для досягнення зазначеної мети дипломної роботи поставлені окремі завдання:

- описати основи технології блокчейн, історію виникнення і розвитку, принципи роботи, консенсус і його основні види, блокчейн-платформи, їх переваги, недоліки та практична цінність;
- зробити аналіз криптовалют, стейблкоїнів, їх класифікація і приклади;
- програмно реалізувати блокчейн-систему з використанням національного стандарту ДСТУ 7564:2014.

Методи дослідження дипломної роботи: аналіз літератури та порівняння.

Об'єктом дослідження є процес дослідження блокчейн в криптовалютній індустрії.

Предметом дослідження є блокчейн, його особливості, типи, принципи роботи та застосування в різних галузях і безпосередньо в криптовалютній індустрії.

Практична цінність отриманих результатів є реалізація блокчейн-системи з використанням національного стандарту ДСТУ 7564:2014, яка є прикладом офіційної інтеграції сфери віртуальних активів у благодійність на державному рівні.

Ключові слова: блокчейн, біткоїн, криптовалюта, консенсус, хешування, цифровий підпис.

ЗМІСТ

ВСТУП

Актуальність. У сучасному світі технологія блокчейн і ринок криптовалюти демонструють динамічний розвиток і передбачають розробку великомасштабних промислових додатків, здатних одночасно керувати багатьма процесами, обробляти і зберігати величезні обсяги даних, забезпечуючи їх логічний взаємозв'язок і узгодженість. Можливості застосування технології блокчейн в бізнесі і промисловості не знають кордонів. Це підтверджується особливою увагою до цих питань з боку державних і приватних структур, високопрофесійних аналітиків та практиків й експертів.

Особливої популярності технологія набула, коли відбувся перший різкий зріст ціни криптовалюти «Біткоін». У той час багато людей почали входити в сферу та зацікавилися новою перспективною технологією, в основі якої була ідея децентралізації, тому що «Біткоін» працює на основі цієї технології. Вагомим внеском у розвиток теоретичних основ блокчейн-технологій стала праця творця блокчейну Сатоші Накамото – «Біткоїн: цифрова пірингова система платежів» [1], а також інші праці таких зарубіжних науковців, практиків та учасників системи блокчейн, як: Натаніель Поппер – книга «Цифрове золото» [2], Майкл Кейсі – «Епоха криптовалют» [3], Ден Шульман – «Блокчейн-революція» [4] та інші. Вони проводили дослідження технології з різних боків, аналіз був досить значущим з самого початку розвитку блокчейну, це сприяло швидкому та ефективному використанню технології іноземними компаніями. Також блокчейн досліджували і вітчизняні фахівці, серед яких А. Усенко та Н. Ющенко, що в наукових працях описували можливі майбутні перспективи застосування технології на території України, що в позитивному сенсі вплине на економіку держави [5].

Метою дипломної роботи є реалізація блокчейн-системи на основі національного стандарту ДСТУ 7564:2014, щоб використати її для благодійності на державному рівні.

Для досягнення зазначеної мети дипломної роботи поставлені окремі завдання:

- описати основи технології блокчейн, історію виникнення і розвитку, принципи роботи, консенсус і його основні види, плюси і мінуси алгоритмів, а також приклади використання, блокчейн-платформи, їх переваги, недоліки та практична цінність;
- зробити аналіз криптовалют, стейблкоїнів, їх класифікація і приклади;
- програмно реалізувати блокчейн-систему з використанням національного стандарту ДСТУ 7564:2014, яка демонструє приклад офіційної інтеграції сфери віртуальних активів у благодійність на державному рівні.

Об’єкт дослідження – це процес дослідження блокчейн в криптовалютній індустрії.

Предмет дослідження – це блокчейн, його особливості, типи, принципи роботи та застосування в різних галузях і безпосередньо в криптовалютній індустрії.

Методи дослідження, що використовуються у дипломній роботі: аналіз та порівняння.

Практична цінність дипломної роботи полягає у реалізації технології блокчейн.

РОЗДІЛ 1
ОСНОВИ ТЕХНОЛОГІЇ БЛОКЧЕЙН

1.1 Поняття блокчейн

Блокчейном називають технологію розподіленої бази даних, яка базується на ланцюжку записів, що зростає на постійній основі. Дана технологія широко використовується в криптовалютах, безпосередньо в технології біткоїн. Таке використання обумовлене тим, що дозволяє виконувати ланцюгові транзакції, що характеризуються швидкістю обробки та високим рівнем захисту від підробки, викрадення і фальсифікації даних [6].

Назва «blockchain» утворена поєднанням двох англійських слів: «block» і «chain» і дослівно перекладається як «ланцюжок блоків». Блок – це файл, який в залежності від системи має певні характеристики: дані, час створення та розмір. Дані у блоках зашифровані за допомогою криптографічних алгоритмів. Ланцюг поєднує блоки у логічну послідовність. Якщо спробувати змінити її, то система відкине такий ланцюжок, оскільки послідовність буде визначено як неправильну. Кожен створений новий блок має посилання на попередній. Такий метод зберігання інформації робить майже неможливим підробку даних. При редагуванні даних у будь-якому блоці зміниться його посилання і таким чином почнеться «ланцюгова реакція» і модифікація буде відхилена. Всі відомості в блокчейні накопичуються і формують постійно оновлювану базу даних, яка є «безмежною» - туди можна записати нескінченну кількість транзакцій [7; 8].

Проста аналогія для розуміння технології блокчейн — це Google Doc. Коли користувач створює документ і ділиться ним з групою людей, він поширюється, а не копіюється чи передається. Це створює децентралізований ланцюг розповсюдження, який надає всім доступ до документа одночасно. Ніхто не заблокований, очікуючи змін від іншої сторони, а всі зміни в документі записуються в режимі реального часу, що робить зміни повністю прозорими [9].

Головне значення використання блокчейну полягає в тому, щоб дозволити користувачам, зокрема тим, які не довіряють один одному, ділитися цінними даними безпечним методом. Щоб запобігти зчитуванню правильної хеш-суми, технологія використовує кілька засобів захисту, серед яких найважливішим є підтвердження роботи та права власності. Звідси випливає те, що учасники транзакцій не можуть обманути один одного, а дані прозорі, оскільки існує єдина база даних [8; 10].

Особливості використання блокчейн [7; 11]:

- розповсюджується в цифровому вигляді на низку комп'ютерів майже в режимі реального часу;
- є децентралізація в тому, що у ланцюгів немає серверів, кожен з користувачів виконує свою функцію та разом підтримують роботу всього блокчейну;
- використовує багатьох користувачів мережі для досягнення консенсусу; застосовує криптографію та цифрові підписи для підтвердження особи;
- прозорість – дані зберігаються у відкритому і загальному доступі;
- для запису нових даних потрібен консенсус вузлів блокчейну. Це дозволяє фільтрувати транзакції і записувати лише легітимні. Замінити хеш нереально;
- теоретично блокчейн можна доповнювати до нескінченності, тому його часто порівнюють із суперкомп’ютером;
- має механізми, які ускладнюють зміну записів;
- має мітку часу і є програмованим.

Переваги технології: однією з основних переваг блокчейнів є рівень безпеки, який він може забезпечити, а це також означає, що блокчейни можуть захищати та захищати конфіденційні дані від онлайн-транзакцій. Для тих, хто шукає швидких і зручних транзакцій, технологія блокчейн також пропонує це. Насправді це займає лише кілька хвилин, тоді як інші методи транзакції можуть тривати кілька днів. Також немає стороннього втручання з боку фінансових установ чи державних організацій, що багато користувачів вважають перевагою.

Недоліки технології: блокчейн і криптографія передбачають використання відкритих і закритих ключів, і, як повідомляється, виникли проблеми з приватними

ключами. Якщо користувач втрачає свій закритий ключ, він стикається з численними проблемами, що робить це одним із недоліків блокчейнів. Ще одним недоліком є обмеження масштабованості, оскільки кількість транзакцій на один вузол обмежена. Через це завершення кількох транзакцій та інших завдань може зайняти кілька годин. Також може бути важко змінити або додати інформацію після її запису, що є ще одним істотним недоліком блокчейну.

1.2 Історія виникнення і розвитку технології блокчейн

У кінці минулого століття однією з важливих проблем було безпечне збереження і передача інтелектуальної власності. Засоби захисту даних, які тоді використовувались, такі як централізовані бази даних, не були надійними і інформація могла бути пошкоджена або викрадена зловмисниками. Саме через це проводилися роботи над покращенням технологій безпечного зберігання та обміну даними [12].

Багато технологій, на яких базується блокчейн, були розроблені задовго до появи біткоїна. Однією з таких технологій є дерево Меркла , назване на честь вченого, фахівця з комп'ютерної техніки Ральфа Меркла. Він описав підхід до розподілу відкритих ключів і цифрових підписів під назвою "tree authentication" у своїй докторській дисертації 1979 року. Згодом він запатентував цю ідею як метод надання цифрових підписів. Дерево Меркла надає структуру даних для перевірки цілісності даних у наборі [13;14].

Ральф Меркл був не єдиним, хто зробив свій вклад у створення основи для технології блокчейн. Криптограф Девід Чаум описав дизайн розподіленої комп'ютерної системи, яка створена і підтримується групою користувачів у його дисертації 1982 року "Комп’ютерні системи, створені, обслуговувані та довірені взаємно підозрілими групами". Система сховища, яку він запропонував, окреслює план досягнення консенсусного стану між вузлами за допомогою зв’язування історії консенсусу в блоки та незмінного позначення часу прив’язаними даними. У системі Чаума кожне сховище підписує, записує та транслює кожну транзакцію, яку

обробляє. У статті також викладено конкретний код для того, щоб реалізувати такий протокол. Чауму також приписують винахід цифрової готівки, і в 1989 році він заснував корпорацію DigiCash [13;15;16].

У 1991 році двоє американських вчених-дослідників, Скотт Сторнетта та Стюарт Габер, опублікували працю, «Як поставити штамп часу на цифровий документ», в якій описали технологію, за допомогою якої не можна було підробити час створення та модифікації електронного документа. Мета полягала в тому, щоб забезпечити повну конфіденційність самого документа, не вимагаючи ведення записів за допомогою служби позначок часу. Для цього використали концепцію криптографічно захищеного ланцюга блоків для зберігання документів. У 1992 році Стюарт Хабер, Скотт Сторнетт і Дейв Байер оновили дизайн, щоб включити дерева Меркла, що дозволило кільком сертифікатам документів перебувати в одному блоці. Пізніше в 1996 році вийшла публікація Андерсона, а в 1997 - Дойла [13;17;18;19].

У 1995-2005 роки було багато інших досліджень, які також допомогли створити блокчейн. Наприклад, з'явилася мережа P2P, концепція, яку популяризував у 1999 році вже неіснуючий Napster. Сервіс допоміг вдихнути життя в мережу P2P, дозволивши побудувати розподілену систему, яка могла б скористатися обчислювальною потужністю та ємністю зберігання тисяч комп’ютерів. Також у ці роки була введена концепція proof-of-work (PoW) для перевірки обчислювальних зусиль і стримування кібератак. Це поступилося місцем Hashcash, алгоритму PoW, який забезпечує заходи боротьби з відмовою в обслуговуванні. Адам Бек, фахівець в області криптографії та шифропанк, ввів Hashcash у 1997 році, щоб обмежити розсилку спаму електронною поштою. У 1998 році комп’ютерний вчений Нік Сабо працює над «бітовим золотом», децентралізованою цифровою валютою. У 2000 році Стефан Конст публікує свою теорію криптографічних захищених ланцюжків, а також ідеї щодо реалізації. Потім, у 2004 році, Хел Фінні представив PoW багаторазового використання. Підхід PoW відіграє важливу роль у видобутку біткоїнів [13;20 - 22].

Компанія Naka-moto, яка є винахідником криптовалюти біткоїна, опублікувала в 2008 році дослідження «Біткоїн: електронна готівкова система однорангових

даних». Автор цього дослідження Сатоші Накамото і вважається, що це псевдонім, який може використовуватися як для окремою людиною так і для групи осіб, які запропонували цю технологію. Згідно з білим документом, інфраструктура блокчейн підтримуватиме безпечні однорангові транзакції без потреби довірених третіх сторін, таких як банки чи уряди. Накамото значно підвищив дизайн, для цього використав метод, що схожий на Hashcash , для визначення часових позначок для блоків, при цьому не вимагається, щоб ці блоки були підписані заздалегідь визначеною довіреною стороною та запровадження параметру складності, що призначений для стабілізації швидкості, з якою блоки додаються до ланцюжка. Фактично, він визначив електронну монету як «ланцюжок цифрових підписів», де кожен з власників передає монету наступному власнику [1;13;23].

У 2009 році Накамото реалізує перший блокчейн як публічну книгу для транзакцій, здійснених за допомогою біткоїна [1;13;24]:

- 3 січня 2009 року Накамото емітував перший блок біткоїнів, підтвердивши концепцію блокчейну. Блок містив 50 біткойнів і був відомий як блок Genesis - він же блок 0;
- 8 січня 2009 року Накамото випустив Bitcoin v0.1 для SourceForge як програмне забезпечення з відкритим вихідним кодом. Зараз Bitcoin доступний на GitHub;
- 12 січня 2009 року відбулася перша транзакція з біткоїнами, коли Накамото надіслав Халу Фінні 10 біткоїнів у блоці 170;
- 12 жовтня 2009 року був створений канал #bitcoin-dev в Internet Relay Chat для розробників біткоїнів;
- 31 жовтня 2009 року створено першу біржу біткоїнів – Bitcoin Market, яка дозволила людям обмінювати паперові гроші на біткоїн;
- 22 листопада 2009 року Накамото запустив форум Bitcointalk для обміну новинами та інформацією, пов’язаними з біткоїнами.

Накамото створив систему так, щоб у ній ніколи не було більше 21 мільйона біткоїнів. Вже емітовано понад 18 мільйонів. Виходячи з поточних темпів майнінгу, біткоїн повинен досягти межі в 21 мільйон приблизно в 2140. Тим часом їх вартість

продовжує зростати, незважаючи на постійні коливання ціни. У жовтні 2009 року біткоїн коштував менше 1 цента. Сьогодні кожен біткоїн коштує понад 35 000 доларів США [13;25].

22 травня 2010 року біткоїн увійшов в історію, коли Ласло Ханьєц заплатив 10000 біткоїнів за дві піци Papa John's . Вартість піци оцінювалася приблизно в 25 доларів США, що тепер коштуватиме понад 350 мільйонів доларів [13;26].

Незабаром програміст на ім'я Джед Маккалеб запустив Mt. Gox, токійську біржу біткоїнів. Mt. Gox було скороченням від «Magic: The Gathering Online eXchange» — назва карткової гри. На піку свого розвитку Mt. Gox обробляла понад 70% усіх транзакцій з біткоїнами. Але в серпні 2010 року хакер скористався помилкою в коді блокчейну і створив понад 184 мільярди біткоїнів у блоці 74 638, заплямувавши репутацію біткоїна. Накамото опублікував нову версію біткоїн-програми, але до кінця року він повністю зник зі сцени біткоїн [13;27].

Незважаючи на зникнення Накамото, траєкторія біткоїна продовжувалася стабільними темпами. До кінця січня 2011 року було видобуто четверту частину ліміту з 21 мільйона біткоїнів. А на початку лютого 2011 року вартість біткоїна дорівнювала долару США. Невдовзі після цього Джед Маккалеб продав Mt. Gox Марку Карпелесу. І незабаром після цього біткоїн досяг паритету з євро та британським фунтом стерлінгів. Пізніше того ж року WikiLeaks почав приймати пожертвування біткоїнами. Однак Mt. Gox було зламано, а біткоїн викрадено, що спричинило штучне падіння вартості та призупинило торгівлю. Потім у жовтні 2011 року був випущений Litecoin, що представляє собою одне з перших допоміжних продуктів біткоїна [13;28].

До 2012 року смак до криптовалюти був добре встановлений. Ціна біткоїна коливалася в районі 5 доларів протягом більшої частини року з багатьма коливаннями вгору і вниз. На початку того ж року Міхай Алісі та Віталік Бутерін запустили журнал Bitcoin Magazine і опублікували свій перший номер у травні. Bitcoin Foundation також був створений для просування біткоїнів та покращення сприйняття громадськості.

Того ж року Coinbase зібрала понад 600000 доларів США у своєму раунді початкового фінансування, що дає змогу стати однією з провідних бірж біткоїн. Крім того, Кріс Ларсен і Джед Маккалеб заснували OpenCoin. Це призвело до розробки протоколу транзакцій Ripple для валютних операцій і платежів у реальному часі і є мережою обміну валют і грошових переказів на основі публічної книги.

Коли настав 2013 рік, біткоїн був добре закріплений і продовжив свою зростаючу траєкторію. У лютому Coinbase повідомила, що продала біткоїнів на 1 мільйон доларів за один місяць за ціною понад 22 доларів кожен. До кінця березня, коли в обігу було 11 мільйонів біткоїнів, загальна вартість валюти перевищила 1 мільярд доларів. А в жовтні того ж року у Ванкувері, Британська Колумбія, запустили перший біткоїн-банкомат.

В кінці 2013 року Таїланд, і Китай заборонили криптовалюти. Федеральний суд США наклав арешт на кошти Mt. Gox в США, а ФБР закрило Silk Road, конфіскувавши 26000 біткойнів [13].

2014 рік став переломним для блокчейну, оскільки фінансові установи та інші галузі почали визнавати та досліджувати його потенціал, переміщаючи фокус з цифрової валюти на розвиток технологій блокчейн. Народжується Blockchain 2.0, що стосується додатків за межами валюти [24].

У 2015 році була створена платформа Ethereum Foundation, який проклав шлях до технології блокчейн, яка буде використовуватися не в криптовалюті. Її ключовою особливістю стали «Smart Contracts» - розумні контракти, які є невеликими програмами, що зберігаються у блокчейні та виконуються при досягненні певних умов. Використовуються для створення домовленостей без використання «посередницьких послуг». Для впровадження «Smart Contracts» потрібна децентралізована мережа з рівними правами для учасників. Основні умови [13;24;29;30]:

- розподілена база даних – це платформа, в якій застосовується смарт-контракт;
- сторони договору, які підтверджують свою участь електронним підписом;

- предмет угоди – це об'єкт, що знаходиться в межах розумного договору або до якого програма має прямий доступ контракту до предмету угоди без людського фактору;
- умови – це алгоритм, що забезпечує коректне виконання всіх частин предмета договору програмною реалізацією.

Того року відбулися й інші важливі події. NASDAQ ініціювала випробування блокчейну. Linux Foundation запустив проект Hyperledger. А дев’ять великих інвестиційних банків об’єднали зусилля, щоб сформувати консорціум R3, досліджуючи, як блокчейн може принести користь. За півроку консорціум розрісся до понад 40 фінансових установ [13].

2016 рік – була використана помилка в коді децентралізованої автономної організації Ethereum, що призвело до хардфорку мережі Ethereum; біржі біткоїнів Bitfinex було зламано і вкрадено майже 120 000 біткоїнів – винагорода вартістю приблизно 66 мільйонів доларів.

2017 рік: Японія визнала біткойн легальною валютою; сім європейських банків сформували консорціум Digital Trade Chain для розробки платформи торгового фінансування на основі блокчейну; компанія Block.one представила операційну систему блокчейн EOS, призначену для підтримки комерційних децентралізованих додатків; приблизно 15% світових банків використовували технологію блокчейн у певній якості.

2018 рік: Південна Корея заборонила анонімну торгівлю криптовалютою, але оголосила, що інвестуватиме мільйони в ініціативи блокчейну; європейська комісія запустила обсерваторію та форум Blockchain; Baidu представила свою платформу blockchain як сервіс.

2019 рік: Walmart запустив систему ланцюга поставок на основі платформи Hyperledger; Amazon оголосила про загальну доступність свого сервісу Amazon Managed Blockchain на AWS; Транзакції мережі Ethereum перевищили 1 мільйон на день; дослідження та розробки блокчейну зайняли центральне місце, оскільки організації скористалися технологією блокчейн і децентралізованими додатками для різноманітних випадків використання.

2020 рік: Ethereum запустив Beacon Chain в рамках підготовки до Ethereum 2.0; Stablecoins значно зросли, оскільки вони обіцяли більшу стабільність, ніж традиційні кібервалюти; зростає інтерес до поєднання блокчейну з штучним інтелектом для оптимізації бізнес-процесів.

Протягом останніх років зростав інтерес до використання блокчейну для інших додатків, ніж кібервалюта. Ця тенденція збережеться і в 2022 році, оскільки уряди та підприємства прагнуть використовувати блокчейн для вирішення різноманітних випадків використання. Це включає голосування, нерухомість, відстеження фізичної форми, інтелектуальні права, інтернет речей та розповсюдження вакцин [13].

1.3 Принципи роботи технології блокчейн

Простіше всього пояснити принцип роботи технології на найстарішим існуючим блокчейном – Bitcoin. Блоки в біткоїн складаються приблизно з 1 МБ даних кожен. Якщо він налічує близько 525000 блоків, тобто приблизно 525000 МБ було збережено в цьому блокчейні. Коли відбувається кожна транзакція, вона записується як «блок» даних. Це утворює великий список усіх транзакцій біткоїн, які коли-небудь відбувалися, аж до першої транзакції. Ці операції показують рух активу, який може бути матеріальним (продукт) або нематеріальним (інтелектуальний) [31;32].

Кожен блок має три основні елементи [9;31]:

- дані в блоці (інформацію на вибір: хто, що, коли, де, скільки і навіть стан — наприклад, температура відправлення їжі);
- 32-розрядне ціле число, яке називається одноразовим. Одноразовий номер генерується випадковим чином, коли створюється блок, який потім генерує хеш заголовка блоку.
- хеш — це 256-бітове число, пов’язане з одноразовим номером. Він повинен починатися з величезної кількості нулів (тобто бути надзвичайно малим).

Розглянемо на прикладі 3 блоків з даними транзакції (рис.1.1). Їх можна порівняти з окремими текстовими документами, які просто описують, які операції відбулися та як вони вплинули на певні баланси. Тоді в «Документ 1» будуть описані хронологічно перші здійснені транзакції з сумарним об'ємом даних до 1 Мб, тоді як наступні транзакції об'ємом від 1 Мб до 2 Мб будуть записані в «Документ 2» тощо. Ці документи і є блоками даних.

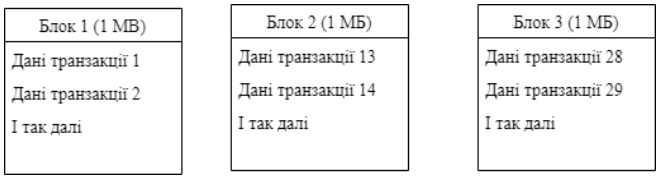

Рисунок 1.1 – Блоки з даними транзакції

Далі блоки пов'язуються разом (тобто - з'єднуються в ланцюжок). Для цього кожен блок отримує унікальний (цифровий) підпис, який точно відповідає рядку даних у цьому блоці. Якщо всередині блоку відбуваються будь-які зміни, навіть якщо змінюється лише один символ, цей блок отримає новий цифровий підпис. Як це працює? Це відбувається за допомогою хешування.

Припустимо, що блок 1 реєструє дві транзакції: «транзакція 1» і «транзакція 2». Уявімо, що дані цих транзакцій займають 1 Мб дискового простору (насправді, зрозуміло, в 1 Мб можна записати набагато більше транзакцій). Цей блок тепер отримує цифровий підпис для цього конкретного рядка даних. Нехай такий підпис буде, наприклад, «5UA» як показано на рис. 1.2:

Рисунок 1.2 – Блок з цифровим підписом

Зміна навіть одного символу у блоці 1 призведе до отримання зовсім іншого підпису. Дані в блоці 1 тепер пов’язані з блоком 2 шляхом додаванням підпису

блоку 1 до даних блоку 2. Підпис блоку 2 тепер частково складається з підпису блоку 1, оскільки він включений в рядок даних у блоці 2 (рис.1.3) [32].

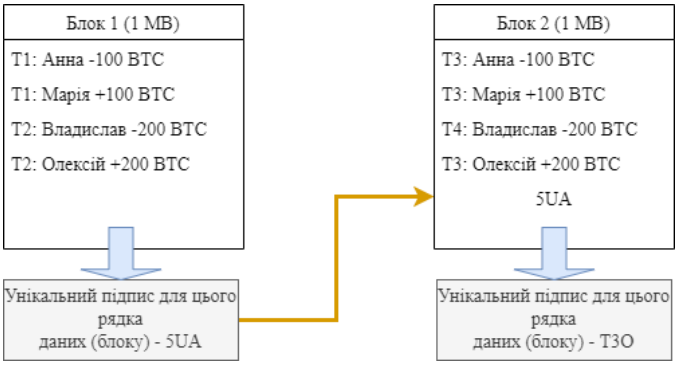

Рисунок 1.3 – Пов’язані 2 блоки

Підписи зв’язують блоки один з одним, утворюючи з них ланцюжок блоків. Три блоки показані на рис. 1.4.

Рисунок 1.4 – Ланцюжок з 3 блоків

Якщо дані в 1 блоці змінити (наприклад транзакція 1 між Анною і Марією: надіслано замість 100 біткоїнів – 500), то рядок даних у блоці 1 тепер інший і також отримує новий підпис, як показано на рис. 1.5, який не відповідає підпису доданому до блоку 2. Перший і другий блок тепер вважаються не зв’язаними один з одним. Це вказує іншим користувачам цього блокчейну на те, що деякі дані в блоці 1 були змінені, і оскільки блокчейн має бути незмінним, вони відхиляють цю зміну, повертаючись до свого попереднього запису блокчейну, де всі блоки все ще з’єднані разом(рис.1.4) [32].

Рисунок 1.5 – Зміна даних у першому блоці

Єдиний спосіб, завдяки якому зміна може залишитися непоміченою, це якщо всі блоки залишаються прив'язаними один до одного. Це означає, що для того, щоб зміна залишилася непоміченою, новий підпис блоку 1 повинен замінити старий у даних блоку 2. Але якщо дані блоку 2 змінюються, це призведе до того, що блок 2 також матиме інший підпис як показано на рис. 1.6. Тепер блок 2 і 3 більше не пов'язані разом.

Рисунок 1.6 – Зміна даних у другому блоці

Блоки в блокчейні є загальнодоступними для всіх. Отже, якщо передбачається, що зміни залишаються непоміченими в блокчейні, усі блоки повинні залишатися належним чином зв'язаними (інакше люди можуть сказати, що певні блоки не пов'язані належним чином один з одним). Це означає, що зміна одного блоку вимагає нового підпису для кожного іншого блоку, який йде після нього аж до кінця ланцюга [32].

Унікальний підпис рядка у блокчейні – це криптографічна хеш-функція, яка цікава тим, що ймовірність того, що ви знайдете два фрагменти даних, що дають

однаковий результат, астрономічно мала. У Bitcoin використовується алгоритм хешування SHA-256, щоб надати блокам свої підписи [32;33].

Розглянемо використання хеш на прикладі першого блоку, де Анна надсилає Марії 100 біткоїнів (рис.1.7)

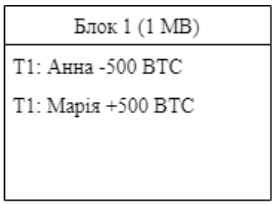

Рисунок 1.7 – Блок 1

Рядок даних з цього блоку виглядає так: «Блок 1 Анна -500 Марія +500». Якщо цей рядок даних вставити в алгоритм хешування, то підпис буде таким: 74a80b2686a34125ddd0bceea10c8c1022cf6066decc002715bb2d6263317c61. Цей хеш додано до даних блоку 2 (Марія передає Владиславу 500 біткоїнів) як показано на рис. 1.8. Рядок даних блоку 2: «Блок 2 Марія – 500 Владислав + 500 74a80b2686a34125ddd0bceea10c8c1022cf6066decc002715bb2d6263317c61», а хеш його: 2860e7b34d4ea3f2782d2c0c66ad09c31cdec4e7a36c28570d41b023b2c8a6af [32].

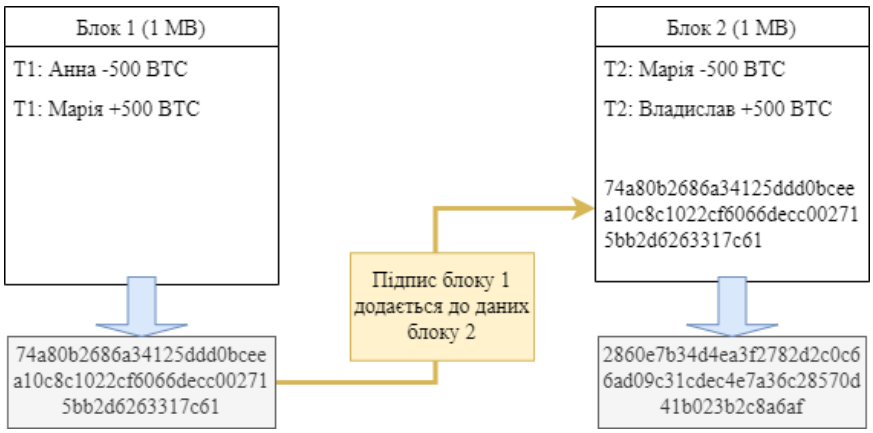

Рисунок 1.8 – Два блоки з’єднані за допомогою хеша

Підпис не завжди відповідає вимогам. Блок буде приєднано до блокчейну лише в тому випадку, якщо його цифровий підпис починається, наприклад, з деякої кількості нулів, що йдуть підряд. Для пояснення припустимо, що тільки блоки з підписом, що починається принаймні з десяти послідовних нулів, підлягають додаванню в блокчейн. Кожен рядок даних має тільки один пов'язаний з нею

унікальний хеш. Що робити, якщо підпис (хеш) блоку не починається з десяти нулів? Для того, щоб знайти в блоку підпис, який відповідає вимогам, рядок даних блоку потрібно багаторазово змінювати поки цей конкретний рядок даних не призведе до підпису, що починається з десяти нулів. Оскільки дані транзакції та метадані (номер блоку, мітка часу тощо) повинні залишатися такими, якими вони є, до кожного блоку додається невеликий спеціальний фрагмент даних, який не має жодної мети, окрім того, щоб знайти відповідний підпис. Цей фрагмент даних блоку називається nonce або код, що одноразово використовується. Nonce — це повністю випадковий рядок чисел. На рис. 1.9 зображено блок, який містить: 1) дані транзакції, 2) підпис попереднього блоку і 3) одноразовий номер. Процес багаторазової зміни nonce та хешування даних блоку для пошуку відповідного підпису називається майнінгом і це те, що роблять майнери. Вони вирішують неймовірно важку математичну задачу, оскільки одноразове значення має лише 32 біти, а хеш — 256, існує приблизно чотири мільярди можливих комбінацій nonce-хеш, які необхідно видобути, перш ніж буде знайдено потрібну. Коли це відбувається, кажуть, що майнери знайшли «золотий одноразовий номер», і їхній блок додається до ланцюжка [32;9].

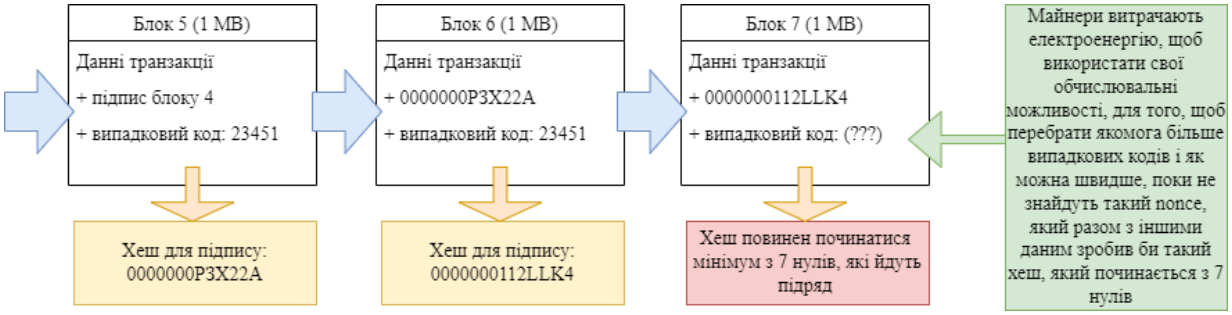

Рисунок 1.9 – Ланцюжок з трьох блоків

Будь-який користувач мережі блокчейну може брати участь у цьому процесі, завантаживши та запустивши відповідне програмне забезпечення для майнінгу для цього конкретного блокчейну. Коли користувач робить це, він просто запускає свою обчислювальну потужність, щоб спробувати знайти одноразовий код для блоку [32].

Однією з найважливіших концепцій технології блокчейн є децентралізація. Жоден комп'ютер чи організація не можуть володіти ланцюжком. Натомість це

розподілена книга через вузли, які підключені до ланцюжка. Взагалі вузол — це точка перетину або з'єднання в телекомунікаційній мережі. В блокчейні вузлом можна вважати будь-який електронний пристрій, який зберігає копії блокчейну та підтримує функціонування мережі. Але незважаючи на те, що всі вони апріорі мають однакові права, вони можуть виконувати різні завдання. Тут багато залежить не тільки від мети власника, а й від конкретних параметрів комп'ютера [9;34;35].

Кожен вузол має власну копію ланцюга блоків, і мережа повинна алгоритмічно схвалити будь-який нещодавно емітований блок, щоб ланцюжок був оновлений. Оскільки блокчейни прозорі, кожну дію в реєстрі можна легко перевірити та переглянути. Кожному учаснику надається унікальний буквено-цифровий ідентифікаційний номер , який показує його транзакції. Основні функції вузлів [9;35]:

- визначення, чи є блок транзакцій легітимним, і приймають чи відхиляють його;
- зберігання блоки транзакцій (зберігання історії транзакцій блокчейну);
- транслювання та поширення історії транзакцій вузлами на інші вузли, яким може знадобитися синхронізуватися з блокчейном (оновлення історії транзакцій важливі).

1.4 Поняття і види консенсусу

Алгоритм консенсусу – це механізм, який дозволяє машинам чи користувачам координувати свої дії в розподіленому середовищі. Його головний обов'язок – це зобов'язатися забезпечити виконання того, що всі агенти в системі можуть домовитися про одне джерело істини, навіть якщо деякі учасники зазнають невдачі. По суті, протокол консенсусу гарантує, що кожен новий блок, який додається до блокчейн, є єдиною версією істини, яка погоджується всіма вузлами в блокчейн [36;37].

Сам процес прийняття рішень також називається алгоритмом консенсусу. В цьому випадку ключовими аспектами є: координація; кооперування; спільна робота;

егалітаризм (рівність усіх голосів); інклюзивність (максимально-можлива кількість людей, що беруть участь у консенсусі); участь [38].

Зараз існує багато алгоритмів консенсусу, які дають різну швидкість обробки та рівень безпеки транзакцій [39].

Proof of Work (перекладається як «доказ роботи») – найстаріший протокол консенсусного алгоритму, який використовується в блокчейні, а також у багатьох криптовалютах, таких як: Litecoin, Ethereum, bitcoin cash, Zcash, Dogecoin та інші. Головна ідея алгоритму полягає в тому, що вузол мережі повинен виконувати роботу майнера, тобто інтенсивні обчислення для підтвердження транзакції і додавання її в пул, а результат можна легко порівняти з іншими підсумками мережевих обчислень. Навіть якщо майнер перепробував трильйони комбінацій, щоб отримати правильний хеш, йому потрібно лише один раз пропустити дані через функцію. Якщо отримані дані дають дійсний хеш, вони будуть прийняті, а майнер, який першим це зробить, отримає винагороду. Інакше мережа відкине його, а користувач даремно витратив час і електроенергію [36;38;40].

Плюси PoW [38]:

- система проста в розумінні;
- алгоритм стійкий до атаки Sybil, коли зловмисник створює підроблені вузли для придушення справжніх чесних користувачів. Однак ймовірність створення підробки дорівнює нулю, оскільки продуктивність обчислювальної техніки є основою підтвердження роботи, а також сполучена з матеріальним світом. З цього випливає, що підтвердження неможливо сфальсифікувати або вкрасти;
- неможливість отримати доказ заздалегідь, оскільки поява кожного нового блоку вимагає запуску обчислювального раунду заново (кожен новий блок містить посилання на попередній);
- алгоритм дає чесний розподіл винагород. На їм розмір впливає хеш-ставка майнера (обчислювальна продуктивність). Майнер завжди отримує винагороду відповідно до отриманим блокам;

- дає мотивацію для чесної і справедливої гри. Оскільки вартість виробництва є матеріальною відносно високою, а нечесна поведінка позбавляє майнера прибутку, користувачам дається додатковий стимул, щоб відмовитися від шкідливих дій.

Мінуси PoW [38;41]:

- має величезні витрати енергії при використанні алгоритму. Система розроблена для забезпечення постійно зростаючої складності завдання, а отже, обчислення вимагає більшої енергії та обчислювальної продуктивності;
- система уразлива до атаки «51%». Об'єкти, які мають доступ до потужніших обчислювальних систем, можуть становити потенційну загрозу для централізації. Зараз 65% видобутку біткоїнів розподіляють п’ять пулів. Ця ситуація є потенційно небезпечною: їхні об’єднані сили можуть призвести до атаки «51%», що може призвести до шкідливої більшості, яка здатна ігнорувати блоки від інших майнерів. Експерти стверджують, що великі системи мають захист від таких ситуацій, тоді як менші блокчейни зі невеликою кількістю учасників все ще мають певні ризики;
- дає можливість створювати вилки(fork). Журнал транзакцій, що оновлюється, обчислюється за допомогою ланцюжка блоків, який містить найбільшу сукупну складність робочих доказів. Згідно з цим, користувачі визначають новий блок за допомогою існуючого, проте не слід забувати про можливість виникнення вилок (вузол або сукупність вузлів, що не погодилися та виходять із групи, надалі продовжуючи самостійно). Через те, що розрахунки виконуються паралельно, вони при сучасних умовах є економічно невигідними;
- низька швидкість процесу транзакцій – означає, що розрахунок займе багато часу, таким чином створюються системи механізмом консенсусу, які не підходять для мікроплатежів або швидких грошових переказів.

Proof-of-Stake (перекладається як «доказ володіння») – це поширений протокол консенсусного алгоритму для обробки транзакцій і створення нових блоків у блокчейні, який використовується в VCash,, NXT, Peercoin, Tezos, BitBay, Stratis та Qtum. Рішення приймається на основі більшого балансу учасника. У кого більша частка криптоактиву(та деякі інші параметри, наприклад, термін її зберігання), той має більше шансів на створення нового блоку. Алгоритм створення блоку не

залежить від потужності комп'ютера. Загалом, чим більше є валюти на рахунку учасника і чим довший період очікування, тим існує більша ймовірність того, що учасник підписав блок і отримає комісійні, які стягуються з транзакційних зборів з блоку [38;42;43].

Плюси PoS [38;43]:

- зменшене споживання енергії порівняно з PoW. Процес алгоритму повністю віртуалізований і не вимагає виконувати великі обчислення;
- щоб заробляти на Proof-of-Stake, не потрібно купувати дороге й потужне обладнання;
- має захист від атаки «51%». Зловмисник не виграє через такий підхід;
- можна заробляти, навіть якщо у користувача мінімум монет – об'єднатися з іншими учасниками для спільного стейкінгу;
- швидко приймаються рішення. Через те, що ключові гравці отримують додаткові голоси, для того, щоб досягти консенсусу потрібно використати менше часу і це прискорює транзакції.

Мінуси PoS [38]:

- є додатковий стимул для децентралізації. Загальний принцип: «чим більша сума валюти, тим вищі потенційні винагороди» дає учасникам мережі додатковий поштовх для збереження валюти. Хоча мережа буде захищена від атак як «51%», є гіпотеза, що якщо 51% коштів буде накопичено в руках однієї людини, то це призведе до подальшого контролю над прийняттям рішень. У великих системах це малоймовірно, тоді як для менших мереж це може бути так;
- більше голосів мають учасники з великими рахунками;
- загроза вилок(fork). У традиційному шаблоні «Доказ володіння» найбільш корисною поведінкою для учасників є створення форків для регулярних витрат. З цієї причини механізм PoS додатково ускладнюється обов'язковими тарифами та іншими умовами, намагаючись закрити «прогалини» в безпеці мережі шляхом введення економічного фактора для стримування нечесних гравців.

Delegated Proof-of-Stake (перекладається як «делеговане підтвердження ставки») – це тип протоколу консенсусу блокчейну, утворений шляхом внесення

змін до PoS, який дозволяє користувачам витрачати свої монети, щоб голосувати за різних делегатів. Після обрання їх вони можуть приймати важливі рішення, які стосуються всієї мережі. Наприклад, обрані делегати можуть встановлювати правила протоколу або перевіряти транзакції. DPoS використовується в EOS, BitShares, TRON і Cosmos [38;44].

Плюси DPoS [38]:

- для всіх учасників мережі важлива чесність в транзакціях. Система репутації є потужним стимулом для підтримки системи у відповідності до правил чесної конкуренції;
- швидкість операцій є вищою, наприклад, порівнюючи з PoS. Учасники можуть співпрацювати, а не конкурувати один з одним, часткова централізація дає змогу швидше досягти консенсусу;
- система не потребує високої обчислювальної продуктивності. Немає потреби перераховувати весь ланцюжок, тому що блок, отриманий з іншого довіреного вузла, що тестується;
- мережі стабільна. Якщо деякі з вузлів виходять з ладу, спільнота може проголосувати, щоб їх замінили.

Мінуси DPoS [38]:

- не знайдено точний ступінь надійності цього алгоритму, але поки що всі можливі намагання зламати його зазнали невдачі;
- власникам гаманців потрібна безперервна мотивація, для того щоб підтримувати свій бізнес на потрібному рівні.

Proof of Concept (перекладається як «доказ ємності») – відправна точку для розробки будь-якого корпоративного блокчейну. У системі увага присвячується простору на жорсткому диску, а не продуктивності обчислень: чим більше місця на ньому, тим більша ймовірність, що майнер знайде потрібний хеш, щоб створити новий блок. Алгоритм має два етапи – побудова графіка, яка створює список на жорсткому диску всіх можливих значень хеш-кодування блоку, розділених на пари, та видалення. Використовується у Burstcoin, BXTB, PermaCoin, SpaceMint [38;45,46].

Плюси PoC:

- головна особливість системи полягає в тому, що алгоритм не потребує спеціального обладнання для роботи, в теорії вузлом може бути навіть звичайний мобільний пристрій з операційною системою Android. Дані можна легко видалити, і користувач далі може використовувати жорсткий диск, щоб виконувати інші завдання.

Мінуси PoC [38]:

- метод може реалізувати алгоритм у невеликих приватних мережах;
- існує загроза для створення шкідливого програмного забезпечення, яке використовує місце на жорсткому диску для розробки без відома учасника. Сама система цих схем не витримає, а для усунення цих перешкод знадобляться окремо створені технології.

Proof-of-authority (перекладається як «доказ повноважень») – це альтернативний механізм консенсусу, який надає невеликій кількості учасників блокчейну повноваження перевіряти транзакції або взаємодії з мережею та оновлювати її більш-менш розподілений реєстр. Згідно з обраною схемою, за генерацію кожного нового блоку транзакцій, які будуть введені до блокчейну, відповідають одна або кілька перевіряючих машин. Новий блок може бути прийнятий безпосередньо без перевірки, або одностайним голосуванням генераторів блоків, або просто більшістю, залежно від конфігурації, обраної для блокчейн. Використовується в Ethereum, Cardano, Apla [38;47,48].

Плюси PoAuthority [38;49]:

- не вимагається високопродуктивне обладнання. У порівнянні з консенсусом PoW, консенсус PoA не вимагає від вузлів витрачати обчислювальні ресурси для вирішення складних математичних завдань, що знижує витрати на обслуговуванні мережі;
- інтервал часу, через який генеруються нові блоки, є передбачуваним. Для консенсусу PoW і PoS цей час відрізняється;

- має високу швидкість транзакцій. Блоки генеруються в певній послідовності через встановлений інтервал часу авторизованими мережевими вузлами. Це збільшує швидкість перевірки транзакцій;
- мережева надійність як достатньо специфічного кола користувачів відповідає за те, щоб прийняти рішення і усунути зловмисників.

Мінуси PoAuthority [38;50]:

- має високий ступінь централізації. Механізм вимагає від користувачів довіряти валідаторам і авторизаторам. З цього випливає те, що використання даного методу може біти доцільним лише для приватних мереж;
- непоширеність алгоритму.

Proof of burn (перекладається як «доказ горіння») – це механізм консенсусу, який використовується для перевірки нових блоків у блокчейні. Цей механізм заснований на тому, що учасники знищують монети: лише ті, хто може довести, що вони вбили заздалегідь визначену кількість монет, вважаються достатньо надійними, щоб взяти на себе перевірку нового блоку. Використовується у Slimcoin [38;51].

Плюси PoB [50;52]:

- процес є економічним за енергією та обладнанням порівняно з іншими механізмами;
- вимагаючи від майнерів спалювати частину своїх монет для створення нових блоків, підвищується безпека системи;
- згоряння монет зменшує оборотну пропозицію (дефіцит ринку);
- заохочує майнерів до довгострокових зобов'язань.

Мінуси PoB [52]:

- PoB не екологічно чистий алгоритм, оскільки спалювані біткоїни генеруються за допомогою майнінгу PoW, що вимагає багато ресурсів;
- не доведено, що він працює у великих масштабах. Щоб підтвердити його ефективність і безпеку, необхідно провести додаткові випробування;
- перевірка роботи шахтарів зазвичай затягується. Це не так швидко, як у блокчейнух Proof of Work;

- процес спалювання монет не завжди прозорий або легкий для перевірений пересічним користувачем.

Byzantine fault tolerance (перекладається як «задача візантійських генералів») – це функція розподіленої мережі, яка має первинний вузол і вторинні вузли. Ці вузли працюють разом, щоб досягти консенсусу. Валідатори/генерали контролюють статус мережі і вони обмінюються повідомленнями і цим запобігають шкідливій поведінці та вибирають правильну версію транзакції. Система використовується у Ripple, Hyperledger, Stellar, Dispatch [38;53;54].

Плюси BFT [38;53]:

- не вимагає значної обчислювальної потужності або використання енергії, оскільки немає жодних майнерів, які вирішують складні рівняння для кожного блоку транзакцій. Це робить його набагато більш екологічним;
- має високу пропускну здатність мережі;
- транзакції не вимагають багаторазового підтвердження;
- має невисокі збори комісій;
- працює використовуючи масштабованість мережі.

Мінуси BFT [38]:

- наявна певна ступінь централізації, що обумовлена делегуванням повноважень з приводу прийняття рішень;
- уразливий до атак Sybil, коли одна сторона може отримати контроль над великою частиною вузлів.

Proof-of-Importance (перекладається як «підтвердження важливості») – це механізм консенсусу, який використовується для визначення того, які користувачі мають право виконувати обчислення, необхідні для додавання нового блоку даних до блокчейну та отримання відповідного платежу. Використовується платформою NEM. PoI винагороджує користувачів, які активно здійснюють транзакції в мережі, спираючись на proof-of-stake. За допомогою PoI вузли повинні надати певну суму валюти, щоб мати право на створення блоків , і вибираються для створення блоку приблизно пропорційно оцінці, яка кількісно визначає їх внесок у мережу [55;56].

Плюси PoI [57]:

- мінімальний вплив на навколишнє середовище;
- не вимагає спеціального обладнання;
- дозволяє майнінг, коли комп'ютер вимкнено.

Мінуси PoI [58]:

- сприяє людям, які накопичують криптовалюту. Це механізм самообмеження, оскільки мета криптовалюти — замінити фактичну валюту. Отже, криптовалюти повинні стимулювати використання, а PoI винагороджують людей за те, що вони тримають велику кількість криптовалют у своїх гаманцях;
- працює на користь багатих інвесторів. Люди, у яких є вільні гроші, щоб тримати монети на своїх рахунках, швидше за все, отримають можливість підробити блок;
- винагороджує інвесторів, які тримають криптовалюту на своєму рахунку протягом коротких періодів часу. Єдине, що має значення, кількість криптовалюти, яка є в гаманці, коли має відбутися підробка.

Можна побачити чітку еволюцію механізмів від оригінальної концепції до швидших і спритніших версій. Але з всіх алгоритмів консенсусу Proof of Work все ще є домінуючим. Більш безпечної та надійної альтернативи досі не запропоновано. Проте існує величезна кількість розробок і досліджень у сфері заміни PoW [36].

1.5 Види блокчейн-платформ, переваги і недоліки, практична цінність

Коли компанія розробляє блокчейн-рішення, щоб задовольнити свої потреби в ланцюжку поставок, неминуче має бути прийняте рішення щодо того, який тип блокчейну найкраще підходить для проекту. Існує чотири основних види блокчейн-платформ: публічні, приватні, консорціуму та гібридні [59].

Публічні блокчейни - це не обмежувальна система розподіленої книги без дозволів. Кожен, хто має доступ до Інтернету, може увійти на платформу блокчейну, щоб стати авторизованим вузлом і стати частиною мережі блокчейну. Вузол або користувач, який є частиною загальнодоступного блокчейну, має дозвіл на доступ до поточних і минулих записів, перевірку транзакцій або підтвердження

роботи для вхідного блоку, а також майнінг. Публічні блокчейни в основному безпечні, якщо користувачі суворо дотримуються правил і методів безпеки. Однак це ризиковано лише тоді, коли учасники не дотримуються протоколів безпеки [60].

Найпоширенішим випадком використання публічних блокчейнів є майнінг та обмін криптовалютами, такими як біткойн. Однак його також можна використовувати для створення фіксованого запису з ланцюгом зберігання, що підлягає аудиту, наприклад, електронне нотаріальне засвідчення публічних записів про право власності на майно. Цей тип блокчейну ідеально підходить для організацій, які засновані на прозорості та довірі, таких як групи соціальної підтримки або неурядові організації. Таким чином, найпоширенішими публічними блокчейнуми є Bitcoin, Ethereum і Litecoin [60;61].

Переваги використання публічних блокчейнів [60;61]:

- повна незалежність від організацій, тому якщо організація, яка його запустила, припинить своє існування, публічним блокчейном все ще можна управляти, доки до нього все ще підключені комп’ютери;
- на відміну від приватного блокчейну, двом вузлам або учасникам не потрібно турбуватися про автентичність іншого. Іншими словами, їм не потрібно особисто знати або довіряти іншим вузлам, оскільки процес підтвердження роботи гарантує відсутність шахрайства в транзакціях. Таким чином, можна сліпо довіряти публічним блокчейнум, не відчуваючи потреби довіряти окремим вузлам;
- відкритість і прозорість мережі. Копія записів блокчейну або цифрової книги доступна на кожному авторизованому вузлі;
- у загальнодоступній мережі може бути стільки учасників або вузлів, що робить її безпечною мережею. Чим більша мережа, тим більший розподіл записів і важче хакерам зламати всю мережу. Поки користувачі ретельно дотримуються протоколів і методів безпеки, загальнодоступні блокчейни здебільшого безпечні.

Недоліки використання публічних блокчейнів [61]:

- процес перевірки роботи дуже енергоємний, оскільки для виконання спеціального алгоритму потрібні спеціалізовані системи (апаратні компоненти);

- мережа може працювати повільно, і компанії не можуть обмежувати доступ або використання. Якщо хакери отримають 51% або більше обчислювальної потужності загальнодоступної мережі блокчейнів, вони можуть в односторонньому порядку змінити її;
- публічні блокчейни погано масштабуються. Мережа сповільнюється, коли до мережі приєднується більше вузлів.

Приватний блокчейн - це блокчейн з обмеженнями або дозволом, що функціонує тільки в закритій мережі. Приватні блокчейни зазвичай використовуються в організації або підприємствах, де лише вибрані члени є учасниками мережі блокчейнів. Рівень безпеки, авторизації, дозволи, доступність знаходиться в руках контролюючої організації. Таким чином, приватні блокчейни схожі на використання загальнодоступних блокчейнів, але мають невелику та обмежену мережу.

Приватні мережі блокчейн розгортаються для голосування, управління ланцюгом поставок, цифрової ідентифікації, володіння активами тощо. Прикладами приватних блокчейнів є: проекти Multichain та Hyperledger (Fabric, Sawtooth), Corda тощо [60].

Переваги використання приватних блокчейнів [60;61]:

- контролююча організація встановлює рівні дозволів, безпеку, авторизації та доступність . Наприклад, організація, яка створює приватну мережу блокчейн, може визначити, які вузли можуть переглядати, додавати або змінювати дані. Це також може перешкодити третім сторонам отримати доступ до певної інформації;
- оскільки вони обмежені за розміром, приватні блокчейни можуть бути дуже швидкими і можуть обробляти транзакції набагато швидше, ніж публічні блокчейни;
- приватні блокчейни досить масштабовані. Тобто можна вибрати розмір свого приватного блокчейну відповідно потреб.

Недоліки використання приватних блокчейнів [61]:

- твердження про те, що вони не є справжніми блокчейнуми, оскільки основна філософія блокчейну — децентралізація;

- важче досягти повної довіри до інформації, оскільки централізовані вузли визначають, що є дійсним;
- не велика кількість вузлів також може означати меншу безпеку. Якщо кілька вузлів виходять з ладу, метод консенсусу може бути скомпрометований;
- вихідний код із приватних блокчейнів часто є приватним і закритим. Користувачі не можуть самостійно перевірити чи підтвердити це, що може призвести до зниження безпеки;
- у приватному блокчейні немає анонімності.

Гібридний блокчейн — це поєднання приватного та публічного блокчейну. Він використовує функції обох типів блокчейнів, тобто можна мати приватну систему на основі дозволів, а також публічну систему без дозволів. Завдяки такій гібридній мережі користувачі можуть контролювати, хто отримує доступ до яких даних, що зберігаються в блокчейні. Лише вибраний розділ даних або записів з блокчейну може бути дозволений для публічного доступу, решта зберігається як конфіденційність у приватній мережі. Гібридна система блокчейну є гнучкою, тому користувачі можуть легко приєднатися до приватного блокчейну з кількома загальнодоступними блокчейнуми. Транзакція в приватній мережі гібридного блокчейну зазвичай перевіряється в цій мережі. Але користувачі також можуть випустити його в загальнодоступний блокчейн для перевірки. Публічні блокчейни збільшують хешування та залучають більше вузлів для перевірки. Прикладом гібридного блокчейну є Dragonchain і IBM Food Trust [59;60].

Гібридний блокчейн має кілька сильних варіантів використання, включаючи нерухомість. Компанії можуть використовувати гібридний блокчейн для приватного запуску систем, але показувати певну інформацію, наприклад списки, для громадськості. Роздрібна торгівля також може впорядкувати свої процеси за допомогою гібридного блокчейну, і добре регульовані ринки, як-от фінансові послуги, також можуть отримати переваги від його використання. Медичні записи можна зберігати в гібридному блокчейні. Запис не можуть переглядати випадкові треті сторони, але користувачі можуть отримати доступ до своєї інформації за допомогою смарт-контракту. Уряди також можуть використовувати його для

приватного зберігання даних громадян, але безпечно обмінюватися інформацією між установами [59;61].

Переваги використання гібридних блокчейнів [61]:

- оскільки він працює в закритій екосистемі, сторонні хакери не можуть здійснити атаку 51% на мережу;
- захищає конфіденційність, але дозволяє спілкуватися з третіми сторонами;
- транзакції дешеві та швидкі;
- пропонує кращу масштабованість, ніж публічна мережа блокчейн.

Недоліки використання гібридних блокчейнів [61]:

- цей тип блокчейну не є повністю прозорим, оскільки інформація може бути захищена;
- оновлення є складним завданням;
- користувачі не мають стимулу брати участь або робити внесок у мережу.

Блокчейн консорціуму — це напівдецентралізований тип, де більше ніж одна організація керує мережею блокчейнів. Це суперечить тому, що є у приватному , яким керує лише одна організація. Більш ніж одна організація може діяти як вузол у цьому типі блокчейну та обмінюватися інформацією або займатися майнінгом. Банківська справа та платежі є двома способами використання цього типу. Різні банки можуть об’єднатися і сформувати консорціум, вирішуючи, які вузли підтвердять транзакції. Дослідницькі організації можуть створити подібну модель, як і організації, які хочуть відстежувати продукти харчування. Він ідеально підходить для ланцюгів поставок, особливо для харчових продуктів та медицини. Прикладами блокчейну консорціуму є: Energy Web Foundation, Global Shipping Business Network Consortium, R3 тощо [59;60;61].

Переваги використання блокчейнів консорціуму [61]:

- більш безпечний, масштабований та ефективний, ніж публічна мережа блокчейнів;
- пропонує контроль доступу.

Недоліки використання блокчейнів консорціуму [61]:

- менш прозорий, ніж публічний блокчейн;

- він все ще може бути скомпрометований, якщо вузол-учасник порушено;
- власні правила блокчейну можуть погіршити функціональність мережі.

Висновки за розділом 1

В першому розділі даної дипломної роботи були розглянуті основні характеристики технології блокчейн. Головними результатами виконання завдань першого розділу є:

Головними результатами виконання завдань першого розділу є:

1) проведено аналіз технології блокчейн та подані основні характеристики;
2) зроблено дослідження історії виникнення і розвитку блокчейн; опис технологій, на яких він базується; подані найвідоміші науковці, які допомогли створенню і видозміненню системи;
3) детально розглянутий принцип роботи технології блокчейн, а також його складових; описаний на прикладі процес транзакції і створення ланцюжків блоків;
4) проведено аналіз консенсусу і його основних видів: Proof-of-Work, Proof-of-Stake, Delegated Proof-of-Stake, Proof of Concept, Proof-of-authority, Proof of burn, Byzantine fault tolerance, Proof-of-Importance з методами роботи, плюсами і мінусами, а також прикладами використання;
5) ретельно досліджені чотири види блокчейн-платформ: публічної, приватної, гібридної і консерціому; проведено аналіз їх переваг і недоліків, способів використання в різних галузях, а також приклади блокчейнів.

РОЗДІЛ 2
ВИКОРИСТАННЯ ТЕХНОЛОГІЇ БЛОКЧЕЙН В КРИПТОВАЛЮТІ

2.1 Поняття і особливості криптовалют

Криптовалютою називають цифрову систему платежів, яка є незалежною від банків, що полягає у відсутності контролю транзакцій. Дана система є одноранговою, що робить можливим надсилання та отримування платежів будь-ким та в будь-якому місці. Наявні кошти не є фізичними грошими, які передаються і обмінюють у реальному світі, криптовалютні платежі наявні лише як цифрові записи до онлайн-бази даних, яка описує конкретні транзакції. Коли ви переказуєте кошти в криптовалюті, транзакції записуються в публічну книгу. Криптовалюта зберігається в цифрових гаманцях.

Свою назву криптовалюта отримала тому, що використовує шифрування для перевірки транзакцій. Це означає, що розширене кодування бере участь у зберіганні та передачі даних криптовалюти між гаманцями та в публічні книги. Метою шифрування є забезпечення безпеки та безпеки [62].

В криптовалютах використовуються обчислення дуже різних цільових функцій. Основними критеріями для них є: можливість будь-кого перевірити їх валідність; невідворотність транзакцій; складність обчислень з прогнозованою швидкістю [63].

Наявні типи криптовалют в загальному можна поділити на дві категорії [64]:

- монети, що містять біткоїни та альткоїни (ці валюти не є біткоїном);
- токени, що є програмними активами, а також наявні у технології блокчейн деяких існуючих платформ.

Хоча криптовалюта працює незалежно і використовує власну платформу, токен — це просто криптовалюта , побудована на основі іншого вже існуючого блокчейну . Усі токени є криптовалютами, але не всі криптовалюти є токенами [65].

Особливості криптотокенів:

- програмовані: токени працюють на програмних протоколах, які складаються із смарт-контрактів , які окреслюють особливості та функції токена та правила взаємодії мережі;
- не мають дозволу: будь-хто може брати участь у системі без потреби в спеціальних облікових даних;
- не мають довіри: жоден центральний орган не контролює систему; замість цього він працює за правилами, визначеними мережевим протоколом;
- прозорість: правила протоколу та його транзакції доступні для перегляду та перевірки для всіх.

Хоча криптотокени, як і криптовалюта, можуть зберігати цінність і обмінюватися, вони також можуть бути розроблені для представлення фізичних активів або більш традиційних цифрових активів, або певної корисності чи послуги. Наприклад, існують криптотокени, які представляють матеріальні активи, такі як нерухомість і мистецтво, а також нематеріальні активи, такі як потужність обробки або простір для зберігання даних. Токени також часто використовуються як механізм управління для голосування за конкретними параметрами, такими як оновлення протоколу та інші рішення, які диктують майбутні напрямки різних проектів блокчейн. Процес створення криптотокенів для виконання цих різних функцій відомий як токенізація [66].

Ринки криптовалют рухаються відповідно до попиту та пропозиції. Однак, оскільки вони децентралізовані, вони, як правило, залишаються вільними від багатьох економічних і політичних проблем, які впливають на традиційні валюти. Хоча навколо криптовалют все ще існує багато невизначеності, наступні фактори можуть мати значний вплив на їх ціни [67]:

- пропозиція: загальна кількість монет і швидкість, з якою вони випущені, знищені або втрачені;
- ринкова капіталізація: вартість усіх існуючих монет і те, як користувачі бачать, що це розвивається;
- преса: те, як криптовалюта зображується в ЗМІ, і наскільки вона висвітлюється;

- інтеграція: ступінь, до якої криптовалюта легко інтегрується в існуючу інфраструктуру, таку як платіжні системи електронної комерції;
- ключові події: основні події, такі як оновлення нормативних документів, порушення безпеки та економічні невдачі.

2.2 Найпоширеніші криптовалюти, їх переваги і недоліки

Bitcoin або біткоїни (BTC) – це консенсусна мережа, яка є новою платіжною системою і повністю цифровими грошима. Біткоїн – це перша однорангова децентралізована платіжна мережа, яка працює від користувачів без центрального органу чи посередника. Зі сторони користувача, біткоїн дуже схожий на готівку для Інтернету. З точки зору користувача, біткоїн — це не що інше, як мобільний додаток або комп'ютерна програма, яка надає особистий біткоїн-гаманець і дозволяє користувачеві відправляти та отримувати біткоїни з ними [68].

Генерація нових монет мережевим вузлом відбувається шляхом вирішення складних однорідних математичних задач (розрахунок хешів SHA-256) у той час, коли він може продемонструвати докази виконаної роботи [69].

Переваги біткоїна [70]:

- ліквідність, доступність та універсальність - це обумовлене тим, що переказ криптовалюти передбачає використання лише кількох хвилин та можна здійснювати оплату різних товарів та послуг... Це полегшує витрачання грошей та їх обмін в країнах світу, при цьому можна отримати бонус у вигляді мінімальної або взагалі нульової комісії. Таку криптовалюту як біткоїни з легкістю можна продати в необхідний момент часу;
- прозорість та анонімність - користувачі біткоїну ідентифікуються за допомогою числового коду та можуть мати декілька відкритих ключів, це не гарантує повну анонімність проте зберігає приватність. У такі системі неможливо відстежити виконані транзакції, що виключає загальнодоступне відстеження користувача. Незважаючи на те, що транзакції можна переглядати постійно, що надає вам прозорість, завдяки технології блокчейн вони є захищеними від різних

видів шахрайства. Крім того, тільки власник гаманця, може знати, скільки у нього біткоїнів;

- незалежність від центральної влади: біткоїн є децентралізованою валютою, тобто вона не регулюється одним урядом або центральним банком. Це означає, що органи влади, швидше за все, не зможуть заморозити і вимагати ваші монети. Також немає життєздатного способу введення оподаткування для біткоїнів. Теоретично це дає користувачам автономію та контроль над їхніми грошима, оскільки ціна не пов'язана з політикою уряду;
- високий потенціал повернення: ціни на біткоїн можуть бути дуже мінливими, різко змінюючи щомісячно і навіть щоденно і це може призвести до високого потенціалу повернення . І зі зростаючою кількістю користувачів, які вважають біткойн перспективною глобальною валютою, багато інвесторів і компаній вирішили прийняти її. Це допомагає збільшити потенціал повернення, особливо для тих, хто купив його за нижчою ціною.

Недоліки біткоїна [71]:

- нестабільність: коли біткоїн був створений Сатоші Накамото, було встановлено обмеження в 21 мільйон біткоїнів, які коли-небудь могли існувати, тому деякі вважають біткойн абсолютно дефіцитним. Цей дефіцит робить біткоїн дуже цінним, але також і те, що робить його ціни змінними, оскільки курс тепер є єдиною змінною, яка може змінитися, щоб забезпечити попит;
- немає державних постанов: інвестування в біткойн не регулюється. Порівнюючи біткоїн та звичну валюту можна сказати, що криптовалютні транзакції не є захищеними юридично та зазвичай не є зворотними, що впливає на їх сприятливість до шахрайства;
- необоротність: через те, що операціям з біткоїнами властива анонімність та нерегульованість, їм притаманна низька безпека. Такі криптовалютні операції неможливо змінити, вони є остаточними та незворотними, це означає, що не можливо нічого зробити у випадку надсилання невірної суми чи у випадку вказання невірного одержувача;

- обмежене використання: незважаючи на те, що зростає кількість компаній, які приймають біткоїн, таких як Microsoft і деякі франшизи Subway, це все ще не є загальноприйнятим. Це обмежує те, куди ви можете витрачати гроші, на відміну від використання кредитної чи дебетової картки.

Litecoin або Лайткоін (LTC) – це це криптовалюта , створена з форка блокчейну біткойн. Її творець - колишній співробітник Google Чарлі Лі. Модифікації, які були внесені в біткоїн, щоб створити блокчейн Litecoin, вимагали лише незначних зусиль з точки зору розробки комп'ютера, причому більшість інновацій надходила від біткоїна. Тим не менш, сила Litecoin полягає в тому, що цих змін небагато, але суттєві [69;71;72]:

- використовується криптографічна хеш-функцію Scrypt, а не SHA-256 як у Bitcoin;
- у чотири рази швидше створення блоків із середнім інтервалом 2,5 хвилини замість 10 хвилин;
- загальна кількість одиниць у чотири рази більше – 84 мільйони замість 21 мільйона;
- складність видобутку, яка змінюється кожні два з половиною дні замість двох тижнів.

Переваги Litecoin [73;74]:

- Litecoin є швидшим і дешевшим варіантом відправки вартості в порівнянні з Bitcoin;
- itecoin є відкритим вихідним кодом , а це означає, що кожен може бачити, як він працює, і будь-хто може вносити зміни в протокол.

Недоліки Litecoin [73]:

- варіант використання Litecoin в криптовалютному просторі залишається вразливим, оскільки він не може відповідати властивостям безпеки та збереження вартості (SoV) Bitcoin або стабільності та швидкості нових стейблкоїн;
- якщо біткойн успішно вирішує проблеми масштабування , ціннісні пропозиції Litecoin як швидшої та дешевшої альтернативи будуть підірвані;

- наплив проектів стейблкоїнів загрожує використанню Litecoin як стабільного засобу обміну;
- централізація багатства вища, ніж у Bitcoin або Bitcoin Cash;
- однією з початкових точок продажу Litecoin був опір ASIC , але з тих пір він відмовився від цієї мети. Тепер він демонструє подібний розподіл хешрейту, як і біткойн, але лише частину безпеки.

Ether (Ethereum) –це відкритий код блокчейн-обчислювальна платформа, яка дозволяє розробникам створювати та використовувати децентралізовані програми. Він децентралізований, тобто не керується централізованою владою. Згідно з CoinDesk, Ethereum прагне змінити те, як працюють програми в Інтернеті сьогодні, надавши користувачам більше контролю, замінивши посередників смарт-контрактами, які автоматично реалізують правила. Ентузіасти Ethereum мають намір повернути контроль користувачам за допомогою блокчейну — технології, яка децентралізує дані, щоб тисячі людей у всьому світі отримували копію. Будь-які централізовані послуги можна децентралізувати за допомогою платформи Ethereum [75].

Особливості Ethereum [76]:

- Ether: криптовалюта Ethereum. Він використовується для оплати обчислювальних ресурсів і комісії за транзакції за будь-яку транзакцію, виконану в мережі Ethereum;
- Smart contracts: проста комп'ютерна програма, яка полегшує обмін будь-якими активами між двома сторонами. Це можуть бути гроші, акції, майно або будь-який інший цифровий актив, який ви хочете обміняти;
- Ethereum Virtual Machine: призначений для роботи як середовище виконання для компіляції та розгортання смарт-контрактів;
- Decentralized applications (Dapps; децентралізована програма): програмне забезпечення, призначене для роботи в мережі Ethereum без контролю централізованої системи: воно забезпечує пряму взаємодію між кінцевими користувачами та децентралізованими постачальниками програм;

- Decentralized autonomous organizations (DAOs; децентралізовані автономні організації): організація, в якій прийняття рішень здійснюється не в руках централізованої влади, а переважно в руках певних призначених органів влади або групи чи призначених людей як частини влади.

Переваги Ethereum [75]:

- наявність смартконтролю;
- розроблюються оновлення(Ethereum 2.0);
- перевірена мережа.

Недоліки Ethereum:

- високі комісії на транзакції;
- низька швидкість транзакцій;
- труднощі під час масштабування;
- застарілий механізм консенсусу.

ADA Carnado – це блокчейн-платформа з proof-of-stake: перша, заснована на рецензованих дослідженнях і розроблена за допомогою методів, заснованих на доказах. Він поєднує в собі передові технології, щоб забезпечити неперевершену безпеку та стійкість для децентралізованих програм, систем і суспільств [77].

Серцем будь-якої блокчейн-платформи є алгоритм, який вона використовує для створення блоків і перевірки транзакцій. Cardano використовує Ouroboros, алгоритм, який використовує протокол proof-of-stake (PoS) для видобутку блоків. Ouroboros був розроблений експертами в області криптографії та інженерії. Він заснований на науково-математичних принципах для підвищення ефективності та безпеки платформи.

Розробка Cardano була унікальною в тому, що вона була заснована на наукових академічних дослідженнях. Кожен із етапів розробки Cardano підтримується дослідницькою основою, що включає рецензовані висновки з методами, заснованими на доказах. Це створює міцну основу для просування вперед у майбутньому як мережі блокчейн, так і токена ADA.

Протокол proof-of-stake, який використовує Cardano, призначений для зменшення витрат енергії під час процесу виробництва блоку. Крім того, цей

протокол включає в себе нескінченно масштабований механізм консенсусу. Ця масштабованість та енергоефективність дозволяють легко майнінг і швидкий час транзакцій на платформі [78].

Переваги Cardano [77]:

- більш екологічно чистий. Cardano — одна з найбільш екологічно чистих блокчейн-систем;
- швидші транзакції. Cardano також набагато швидше обробляє транзакції, ніж Bitcoin або Ethereum 1.0;
- рецензована мережа.

Недоліки Cardano [77;78];

- наздоганяючи більш відомих конкурентів. Cardano намагається створити кращу версію блокчейну, але конкуренти, такі як Ethereum, мають перевагу в більш тривалій історії використання та більшої популярності розробниками. Насправді, одне з оновлень Ethereum 2.0 включає підхід з доказом ставок, який може звести нанівець ключову перевагу Cardano;
- популярність. Ринок криптовалют стає все більш переповненим, і привернути увагу нелегко. Наприклад, Dogecoin показав, наскільки криптовалюта може процвітати лише на основі популярного мему, якого немає у більш стриманого бренду Cardano.

2.3 Поняття стейблкоїнів, їх класифікація і приклади

Стейблкоїн — це цифрова валюта, яка прив’язана до «стабільного» резервного активу, такого як долар США або золото. Стейблкоїни призначені для зниження волатильності відносно незалежних криптовалют, таких як біткойн [79].

Принцип використання монет схожий на те, як використовують фізичні гроші. Насправді вартість монет майже завжди дорівнює активам у співвідношенні 1:1, а самі стейблкоїни дозволяють заощадити капітал, провести обмін або розрахунок [80].

Практична цінність [79]:

- мінімізування волотивності: актив, прив’язаний до більш стабільної валюти, може дати покупцям і продавцям впевненість у тому, що вартість їхніх токенів не підвищиться або не впаде непередбачувано в найближчому майбутньому;
- розсилання і зберігання активів: для збереження стейблкоїнів не потрібен банківський рахунок, і їх легко перевести, стейблкоїни можна легко розсилати по всьому світу, в тому числі в місця, де долар США важко отримати або місцева валюта нестабільна;
- зароблення відсотків;
- переказання грошей з маленькою комісією.

Стейблкоїни за ознакою стаблізації вартості можна класифікувати на такі основні групи [81;82]:

- фіатно-забезпечені стейблкоїни(USDT, TUSD, PAX, BUSD) – для забезпечення використовують резерв фіатної валюти;
- крипто-забезпечені стейблкоїни(Bitshares USD, DAI) – забезпечені іншими криптовалютами і мають надмірну заставу, тобто вартість криптовалюти, яка зберігається в резервах, перевищує вартість випущених стейблкоїна;
- алгоритмічні стейблкоїни(Carbon) – можуть утримувати резервні активи, а можуть і не мати. Їхня головна відмінність — це стратегія підтримки стабільної вартості стейблкоїна шляхом контролю його пропозиції за допомогою алгоритму , по суті, комп’ютерної програми, що виконує задану формулу.

Найпопулярніші стеблоїни [80]:

- Tether – валюта, забезпечена фіатними грошима, винайдена для поєднання крипти та стійких валют;
- Тіберій – це швейцарські монети, стабільність яких забезпечують ціни на сім дорогоцінних металів;
- TrueUSD – монета, прикріплена до долара США на основі ефіру;
- Gemini Dollar — це стабільна монета, яка регулюється Державним департаментом США, і її вартість також прив’язана до долара США;
- Candy — ліцензований монгольський стейблкойн.

Висновки за розділом 2

В другому розділі даної дипломної роботи було розглянуто , наведено характеристики та інші важливі відомості для досягнення мети дипломної роботи. Головними результатами виконання завдань другого розділу є:

1) проведено аналіз криптовалют та подані основні характеристики;

2) розглянуто найпопулярніші зараз криптовалюти, а також їх переваги і недоліки;

3) зроблено дослідження стейблкоїнів, їх класифікація і приклади.

РОЗДІЛ 3
ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Програмне середовище для реалізації програмного продукту

Для написання програми використаємо мову C++, тому що вона має дуже гарне поєднання ефективної міжпотокової комунікації та оптимізації однопоточної продуктивності. Оригінальний код для Bitcoin також був написаний на C++.

Особливості C++ [83]:

- контроль пам’яті: блокчейн повинен взаємодіяти з великою кількістю ненадійних кінцевих точок, одночасно забезпечуючи швидке обслуговування будь-яких і всіх вузлів. Щоб задовольнити всі ці вимоги та працювати на найвищому рівні, вам потрібен жорсткий і повний контроль над ЦП і пам’яттю. використання. C++ надає це своїм користувачам;
- потоки: одна з головних проблем програмування на блокчейні — це інтеграція завдань, які добре паралелізуються, і завдань, які не розпаралелюються. Більшість мов спеціалізуються на одній, однак здатність C++ до потоків достатньо хороша для вирішення як паралельних, так і непаралельних завдань;
- семантика переміщення: семантика переміщення забезпечує спосіб переміщення вмісту між об’єктами, а не копіювання безпосередньо;
- поліморфізм часу компіляції: Використовуючи поліморфізм, ви використовуєте певну функцію кількома способами. Подробиці тут;
- ізоляція коду: C++ має функції простору імен, які можна імпортувати з однієї програми в іншу. Простір імен допомагає уникнути колізій імен. Крім того, оскільки C++ має класи, він може діяти як межі між різними API та допомагати в чіткому розділенні.

Переваги C++ для створення блокчейн-додатків включають в себе те, що C++ дуже портативна мова для розробки додатків на кількох пристроях, багатоплатформенна, є потужною, ефективною, швидкою та багаторазовою, що

дозволяє використовувати письмовий код більше одного разу. Ця мова програмування забезпечує продуктивність, ефективність пам’яті та повний контроль. Це також об’єктно-орієнтована програма, яка включає класи, успадкування, поліморфізм, абстракцію даних та інкапсуляцію. Окрім програмування на високому рівні, також можна програмувати на низькому рівні тією ж мовою.

Недоліки C++ для побудови блокчейн-додатків включають в себе те, що використання цієї мови може стати складним у дуже великій програмі високого рівня. Це може ускладнити розуміння, оскільки вона містить складні інструкції або функції, які може бути важко реалізувати, якщо користувач не повністю розумієте програму. Вона також не підтримує сміттєву програму і займає багато часу для написання програми.

Проте, незважаючи на ці незначні недоліки, мова програмування залишається популярною для додатків на основі блокчейну [84].

Для реалізації блокчейн мовою C++ використано програмне середовище від корпорації Microsoft, що являє собою інтегроване середовище розробки переважно використовується для програмного забезпечення,а саме Visual Studio 2019.

Visual Studio, створена компанією Microsoft, є інтегрованим середовищем розробки для створення графічного інтерфейсу користувача, веб-додатків, веб-програм, консолі, хмарних програм та інше. З допомогою цього програмного середовища можна створювати як керований так і рідний програмний код. Для цього Visual Studio використовує різноманітні платформи програмного забезпечення, наприклад Windows Store, Microsoft Silverlight, Windows API тощо. Це програмне середовище підтримує різні мови програмування для написання коду, наприклад, #, C++, VB (Visual Basic), Python, JavaScript та інші. В загальному Visual Studio підтримує 36 відомих мов програмування. Ця програмна платформа доступна для користувачів Windows та macOS [85].

3.2 Функція хешування «Купина» за стандартом ДСТУ 7564:2014

Як розглянуто, у першому розділі дипломної роботи, де досліджена структура блокчейну: блок у ланцюгу обов'язково повинен мати цифровий підпис, який можна створити за допомогою хеш-функції. Наприклад, блокчейн Bitcoin використовує алгоритм хешування SHA-256 (Secure Hash Algorithm). Зараз існує багато хеш-функцій, які можна використати в блокчейні, але розглянемо хешування з національного стандарту України ДСТУ 7564:2014 і порівняємо з іншими функціями [86].

Купина – це функція хешування, що заснована на архітектурі Меркле-Дамгора і забезпечує високу стійкість та гнучке криптографічне перетворення, яке описане в стандарті ДСТУ 7564:2014 «Інформаційні технології. Криптографічний захист інформації. Функція хешування». Цей стандарт хешування забезпечує цілісність та перетворення, що використовується для цифрового підпису. Хеш-функція Купина використовує функцію стиснення Девіса-Мейєра, що побудована за допомогою схеми блокового шифру Івена-Мансура. Основними режимами роботи хеш-функції з ДСТУ 7564:2014 є «Купина-256», «Купина-384» і «Купина-512» [87].

Хеш-функція побудована за принципом шифру Калина, це означає, що функція заснована на функції стиснення, тобто складається з двох фіксованих перестановок. Структура хеш-функції є подібною до алгоритму AES, що означає підстановлювально-перестановочну мережу та в результаті виконання є послідовність бітів довжиною від 8 до 512 біт.

Ключовою відмінністю Купини від інших хеш-функцій є те, що вона може використовуватись в технологіях блокчейн, які враховують динамічність функції і довжину виходу хешу. Під час використання не статичного розміру блоку динамічність виходу функцій хешування не має впливу на кількість транзакцій, які входять у блок, але покращує криптостійкість і інші криптографічні властивості блокчейну [86].

Якщо порівнювати Купину з іншими хеш-функціями, то перше, що можна сказати – це довжини виходу, наприклад, SHA-256 має тільки 4 можливі значення:

224, 256, 384 і 512 біт, а Купина передбачає 64 довжини: 8, 16, … , 256, 264, … , 504, 512. Більш розгорнуте порівняння хеш-функцій представлене в таблиці 3.1 [86].

Таблиця 3.1

Порівняльний аналіз функцій хешування

Функція хешування	Максимальний розмір повідомлення	Довжина значення хешу (біт)	Швидкість шифрування (Мбіт/с)	Розмір блоку	Кількість раундів	Розмір слова	Стійкість до методів криптоаналізу
Купина–512	$< 2^{96}$	512	3,25	1024	14	64	+
MD4	$< 2^{64}$	128	2,36	512	48	32	-
SHA-2/256	$< 2^{64}$	256	1,85	512	64	32	+
SHA-2/512	$< 2^{128}$	1024	1,76	1024	80	64	+
Snerfu	$< 2^{64}$	128	1,70	512	64	128	-

Характеристика конструкції функції хешування Купина: до входу хеш-функції дається повідомлення M – це послідовність бітів N-довжини. Після цього воно доповнюється по певним правилам, щоб була довжина, яка кратна розміру блоку і поділяється на блоки: $m_1, \dots, m_k$, які мають довжину l-біт кожен (l вираховується відповідно до розміру хеш-значення $n, n \in \{8 * s | s = 1,2, \dots ,64\}$:

$$l = \begin{cases} 512, \text{якщо } 8 \le n \le 256 \\ 1024, \text{якщо } 256 < n \le 512 \end{cases} . \quad \#(3.1)$$

Обчислення значення хешу відбувається по такій ітеративній процедурі [86]:

$$h_0 = IV, \#(3.2)$$

$$h_v = T_1^{\oplus}(h_{v-1} \oplus m_v) \oplus T_l^{+}(m_v) \oplus h_{v-1}, v = 1, 2, \dots , k, \#(3.3)$$

$$H(IV, M) = R_{l,n}\left(T_l^{\oplus}(h_k) \oplus h_k\right), \#(3.4)$$

де *IV* – вектор ініціалізації довжиною *l* біт,

$T_l^{\oplus}$, T_l^{+} – це бієктивні перетворення, що відображають вхідний блок, який має довжину *l* біт у вихідний з такою самою довжиною,

$R_{l,n}$ – функція, що повертає *n*-бітів із вхідного блоку *x* довжиною *l* біт (*n*>1), де результат записується в молодші *n*-біти обчисленого значення.

На рис. 3.1 зображена структурна схема функції хешування Купина в загальному вигляді [86].

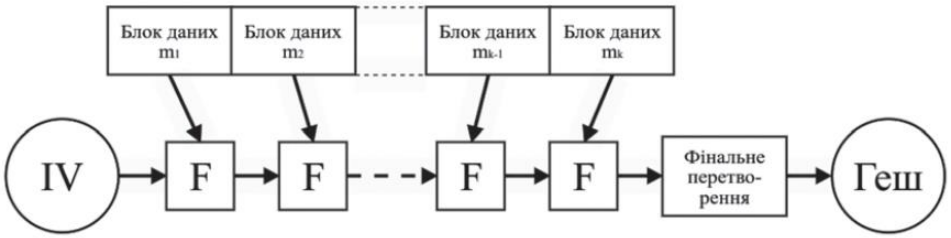

Рисунок 3.1 – Загальна структурна схема хеш-функції Купина

Як можна побачити по схемі: функція стиснення F використовує дві перестановки $T_l^{\oplus}$ та T_l^{+} і обчислюється за формулою:

$$F(m,h) = T_l^{\oplus}(h \oplus m) \oplus T_l^{+}(m) \oplus h\,. \#(3.5)$$

Тоді на *k*-му кроці:

$$h_k = F(m_k, h_{k-1})\,, \#(3.6)$$

Графічне представлення структури функції стиснення в алгоритмі Купина наведено на рис. 3.2.

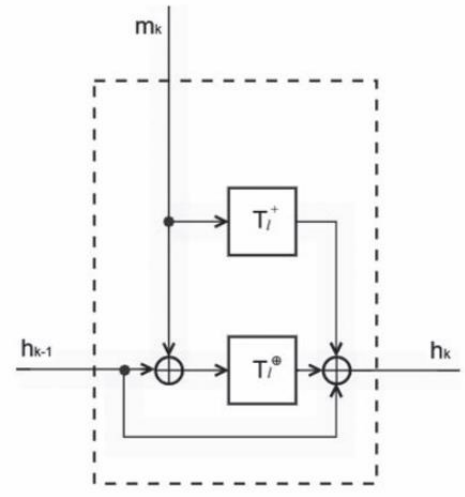

Рисунок 3.2 – Структура функції стиснення

3.3 Аналіз програмної реалізації блокчейну

Програмна реалізація представляє собою реалізацію технології блокчейн за допомогою мови програмування C++. Вихідний код програми представлений в Додатку А. Ця програмна реалізація призначена для вивчення властивостей, розуміння принципу роботи та аналізу технології блокчейн в криптавалюті.

Як відомо, з попередніх розділів дипломної роботи, де детально було розглянуто принцип роботи технології: блокчейн складається з серії блоків, які містять дані, і кожен блок містить криптографічне представлення попереднього блоку, що означає, що стає дуже важко змінити вміст будь-якого блоку без необхідності змінювати кожен наступний; отже, де блокчейн по суті отримує свої незмінні властивості.

Створюємо клас блоку, за допомого рядків, які зображені на рис. 3.3, у новому заголовному файлі Block.h.

```
class CBlock final {
public:
    string PrevHash;
    CBlock(std::uint32_t indexIn, const string & dataIn);
    void SetAsGenesis();
    uint32_t GetIndex() const;
    time_t GetTime() const;
    string GetHash() const;
    void DOMine(uint32_t difficulty);
private:
    uint32_t m_index;
    int64_t m_nonce;
    string m_data;
    string m_hash;
    time_t m_time;
    const string CalculateHash() const;
};
```

Рисунок 3.3 – Клас блоку

Програма викликає клас CBlock (рядок 1), за яким слідує публічний модифікатор (рядок 2) і публічна змінна PrevHash на 3 рядку (щоб кожен блок пов’язати з попереднім блоком). Сигнатура конструктора (рядок 4) приймає параметри indexIn і dataIn ; ключове слово const використовується разом із модифікатором посилання (&), щоб параметри передавалися за посиланням, але не могли бути змінені, це робиться для підвищення ефективності та економії пам’яті. Далі вказується індекс(рядок 6), мітка часу (рядок 7) і цифровий підпис блоку (рядок

8), а потім підпис методу DOMine – рядок 9, який приймає параметр difficulty. Вказуємо приватний модифікатор – рядок 10, за яким слідують приватні змінні на рядках 12–16. Підпис для CalculateHash на 17 рядку також містить ключове слово const, щоб переконатися, що метод не може змінити жодну зі змінних у класі блоків, що дуже корисно під час роботи з блокчейном.

Створюємо новий заголовний файл BlockChain.h, в якому посилаємося на Block.h, і там викликаємо клас блокчейну як зображено на рис. 3.4. На 2 рядку аналогічно є публічний ідентифікатор, за яким слідує підпис конструктора(рядок 3). Далі йдуть три сигнатури (рядки 4-6) і одна з них (рядок 5) приймає параметр newBlock, який є об'єктом класу CBlock. На 8 рядку вказується приватний модифікатор, за яким слідують дві приватні змінні (рядки 9-10), а також сигнатури методів (рядки 11-12), біля яких є слово const , щоб позначити, що вихід методу не можна змінити.

```
class CBlockChain {
public:
    CBlockChain();
    void AddGenesis();
    void AddBlock(CBlock newBlock);
    void PrintBlocks();

private:
    uint32_t m_difficulty;
    vector<CBlock> m_chain;
    const CBlock GetLastBlock() const;
    const uint32_t GetChainSize() const;
};
```

Рисунок 3.4 – Клас блокчейну

У заголовному файлі Kupyna.h і вихідному файлі Kupyna.cpp реалізуємо техніку хешування Купина за національним стандартом ДСТУ 7564:2014, щоб створити хеши(підписи) для блоків.

Створюємо вихідний файл Block.cpp для блоку і реалізуємо конструктор блоків як показано на рис. 3.5. Конструктор починається з повторення підпису, який вказано у файлі Block.h, але також додаємо код для копіювання вмісту параметрів у змінні m_index та m_data. Змінна m_nonce має значення -1 (рядок 2), а змінна m_time — поточний час (рядок 3).

```
CBlock::CBlock(uint32_t indexIn, const string & dataIn) : m_index(indexIn), m_data(dataIn) {
    this->m_nonce = -1;
    this->m_time = time(nullptr);
}
```

Рисунок 3.5 – Конструктор блоків

Добавимо засоби доступу до індексу, мітки часу і хешу блоку як показано на рис. 3.6. Вказуємо підпис для GetHash (рядок 7), а потім додаємо повернення для приватної змінної m_hash (рядок 8), з іншими даними аналогічно.

```
uint32_t CBlock::GetIndex() const {
    return this->m_index;
}
time_t CBlock::GetTime() const {
    return this->m_time;
}
string CBlock::GetHash() const {
    return this->m_hash;
}
```

Рисунок 3.6 – Засоби доступу до даних

Як було розглянуто в попередніх розділах: для успішного створення дійсного блоку майнер повинен створити криптографічний хеш блоку, який він хоче додати до блокчейну, який відповідає вимогам для дійсного хешу на той момент; це досягається шляхом підрахунку кількості нулів на початку хешу, якщо кількість нулів дорівнює або перевищує рівень складності, встановлений мережею, цей блок є дійсним. Якщо хеш недійсний, змінна під назвою nonce збільшується, і хеш створюється знову; цей процес, який називається Proof of Work (PoW), повторюється до тих пір, поки не буде отримано дійсний хеш.

Для цього додаємо метод DOMine як показано на рис. 3.7.

```
void CBlock::DOMine(uint32_t difficulty) {
    char * cstr = new char[difficulty + 1];
    for (uint32_t i = 0; i < difficulty; ++i) {
        cstr[i] = '0';
    }
    cstr[difficulty] = '\0';
    string str(cstr);
    do {
        this->m_nonce++;
        this->m_hash = CalculateHash();
    } while (!str.compare(this->m_hash.substr(0, difficulty)));
    cout << "Block mined: " << this->m_hash << endl;
}
```

Рисунок 3.7 – Використання Proof of Work

У 1 рядку знаходиться сигнатура для методу DOMine, яка була вказана у заголовному файлі Block.h. Далі (2 рядок) створюємо масив символів довжиною на один знак більше, ніж значення для difficulty. Цикл for (рядки 3-5) використовується для заповнення масиву нулями, після чого кінцевому елементу масиву надається символ закінчення рядка (\0) – рядок 6, потім масив символів перетворюється на стандартний рядок (рядок 7). Після цього використовуємо цикл do…while (рядки 8-11) для збільшення значення m_nonce, а m_hash призначається вивід CalculateHash, перша частина хешу потім порівнюється з рядком нулів, який щойно був створений і якщо збіг не знайдено, то цикл повторюється, поки відповідність не буде знайдена. Як тільки це станеться, то у вихідний буфер надсилається повідомлення про те, що блок успішно видобуто (рядок 12).

Створюємо метод CalculateHash як показано на рис. 3.8.

```
inline const string CBlock::CalculateHash() const {
    stringstream ss;
    string s1;

    kupyna_t ctx;
    uint8_t hash_code[512 / 8];
    ss << this->m_index << this->m_time << this->m_data << this->m_nonce << this->PrevHash;
    KupynaInit(200, &ctx);
    uint8_t test[256];
    s1 = ss.str();
    std::memcpy(test, s1.data(), s1.size());
    KupynaHash(&ctx, test, 512, hash_code);
    string sq((char*)(256,hash_code));
    return sq;
}
```

Рисунок 3.8 – Метод CalculateHash

Використаємо підпис методу у рядку 1, який був вказано у файлі Block.h, при цьому додавши ключове слово inline, яке робить код більш ефективним, оскільки компілятор розміщує інструкції методу всередині, де б він не був викликаний. Як наслідок – це зменшує кількість викликів окремих методів. Потім створюємо рядковий потік (рядок 2), за яким до потоку (рядок 7) додаються значення для m_index, m_time, m_data, m_nonce та PrevHash. Завершуємо, повертаючи вихідні дані методу KupynaHash (з файлів Kupyna.h і Kupyna.cpp, які додали раніше), використовуючи вихідні дані з потоку рядків (рядок 12).

Створюємо вихідний файл BlockChain.cpp для блокчейну, реалізуємо конструктор як показано на рис. 3.9.

```
CBlockChain::CBlockChain() {
    this->m_difficulty = 10;
}
```

Рисунок 3.9 – Реалізація конструктора блокчейну

В рядку 1 вказуємо підпис, який використали у файлі BlockChain.h. Далі встановлюємо рівень difficulty, від якого буде залежить важкість процесу PoW.

Далі створимо код для додавання блоку в блокчейн як показано на рис. 3.10. В рядку 1 вказуємо сигнатуру, яка була у файлі BlockChain.h для AddBlock. Потім встановлюємо змінну PrevHash для нового блоку з хешу останнього блоку в блокчейні, який отримуємо за допомогою GetLastBlock та його методу GetHash (рядок 2). Потім блок видобувається за допомогою методу DOMine (рядок 3), а потім блок додається до вектору m_chain (рядок 4), таким чином завершуючи процес додавання блоку до блокчейну.

```
void CBlockChain::AddBlock(CBlock newBlock) {
    newBlock.PrevHash = GetLastBlock().GetHash();
    newBlock.DOMine(this->m_difficulty);
    this->m_chain.push_back(newBlock);
}
```

Рисунок 3.10 – Додавання блоку в блокчейн

Завершуємо файл BlockChain.cpp, додавши останні методи з BlockChain.h – це GetLastBlock і GetChainSize (рис.3.11). Перший використовується для повернення останнього блоку, знайденого у векторі m_chain, а другий для виведення розміру m_chain.

```
const CBlock CBlockChain::GetLastBlock() const {
    return this->m_chain.back();
}

const uint32_t CBlockChain::GetChainSize() const {
    return this->m_chain.size();
}
```

Рисунок 3.11 – Методи для повернення даних з вектора m_chain

У файл main.cpp додаємо наступні рядки, які показані на рис. 3.12. У 2 рядку програма створює новий блокчейн, а далі повідомляє користувача, що блок видобувається шляхом друку у вихідний буфер (рядок 4), а потім створює новий блок і додає його в ланцюжок (рядок 5); процес видобутку цього блоку почнеться,

поки не буде знайдено дійсний хеш. Після видобутку блоку процес повторюється ще для двох блоків.

```
int main(int argc, char* argv[]) {
    CBlockChain bChain = CBlockChain();

    cout << "Genesis..." << endl;
    bChain.AddGenesis();

    cout << "Mining block 1..." << endl;
    bChain.AddBlock(CBlock(1, "Block 1 Data"));

    cout << "Mining block 2..." << endl;
    bChain.AddBlock(CBlock(2, "Block 2 Data"));

    cout << "Mining block 3..." << endl;
    bChain.AddBlock(CBlock(3, "Block 3 Data"));

    bChain.PrintBlocks();

    system("pause");

    return 0;
}
```

Рисунок 3.12 – Основний метод

Для перевірки справності виконання алгоритму запускаємо створену програму (рис. 3.13)

```
C:\Users\alina\source\repos\ConsoleApplication5\Debug\Console...
Genesis...
Mining block 1...    69716CEE81EFC23A0856B05AFD6CCBBC
    2FBE0C7797AABD916CD8D35FB436CE79

Mining block 2...    3E7173605328O4A87C9EA0AD7519B7D5
    B0FABB254D47AA197FA2CCB20D791837

Mining block 3...    35532338DD6065E9A69950B1D6FC9B4A
    E3E3F1202AD8F8EC859D5E7DB36CF9DA
```

Рисунок 3.13 – Результат виконання алгоритму

3.4 Використання системи блокчейну для благодійності на державному рівні

Після введення національного стандарту Купина, його досліджували в Канаді, США, Австрії та інших країнах, були отримані результати, що підтверджують стійкість криптографічного перетворення. ДСТУ 7564 був включений до складу програмних бібліотек. Після цього розглядалося питання впровадження функції хешування Купина у технології блокчейн з метою розроблення державних систем блокчейну, придатних для захисту інформації у межах України.

Після початку повномасштабної війни в Україні благодійність перейшла на новий рівень. Багато приватних фондів з’явилося, так само як і державні отримали свій розвиток. Банки України зайняли важливу нішу у цьому всьому. Можна сказати, що робиться все, щоб спростити збирання коштів для військових і постраждалих від війни. Але не треба забувати, що ще важлива захищеність транзакцій і коштів.

Банки мають дуже багато недоліків: ліміти транзакцій, технічні збої в додатках і взагалі на серверах, часто через перегруженість, підпорядкованість одній людині та інші. А криптовалюта працює без банків і бюрократії. Зараз швидкість допомоги має велике значення. Приватні благодійні фонди вже стикнулися з цими проблемами і почали використовувати криптовалютні гаманці. Кошти в криптовалютах дістаються до України набагато швидше, ніж допомога ООН. І є можливості купляти допомогу в криптовалюті.

Блокчейн поступово змінює усю парадигму в роботі благодійних організацій завдяки своїм основним перевагам: децентралізованість, прозорість та беззмінне зберігання записів. Завдяки технології блокчейн та фінтех є можливість для створення в Україні глобальної державної екосистеми для благодійності. Дуже вдалим рішенням при цьому використати у системі хеш-функцію з національного стандарту ДСТУ 7564, яка має свої переваги і перевіреність.

Висновки за розділом 3

В третьому розділі даної дипломної роботи було розроблено і описано програмну реалізацію блокчейну. Головними результатами виконання завдань третього розділу є:

- вибір програмного середовища для реалізації застосунку, цим продуктом є Visual Studio 2019 від розробника Microsoft;
- вибір мови програмування для реалізації блокчейну, цією мовою програмування є мова C++;

- дослідження функції хешування «Купина» за стандартом ДСТУ 7564:2014 та порівняння її з іншими хеш-функціями;
- детальний аналіз програмного продукту з використанням хеш-функції Купина;
- перевірка працездатності програми;
- аналіз практичної цінності створеної системи, як прикладу глобальної державної екосистеми для благодійності.

ВИСНОВКИ

Під час написання дипломної роботи було досліджено технологію блокчейн в криптовалютній індустрії і як з її допомогою можна покращити економіку України, а також допомагати військовим, що зараз, під час повномасштабної війни, є найбільш актуальним.

У першому розділі було описано основи технології блокчейн, історію її виникнення і розвитку. Потім розглянуто структуру, принципи роботи та особливості. Далі досліджено таке поняття як консенсус і його основні види, плюси і мінуси алгоритмів, а також приклади використання. В кінці розділу було проаналізовано основні види блокчейн-платформ: публічна, приватна, гібридна і консорціум, їх переваги, недоліки та практична цінність.

У другому розділі було проаналізовано криптовалюти та подані їх основні характеристики. Далі розглянуто найпопулярніші зараз криптовалюти, а також їх переваги і недоліки. В кінці зроблено дослідження стейблкоїнів, їх класифікація і приклади.

У третьому і останньому розділі дипломної роботи було реалізовано блокчейн-систему з використанням національного стандарту ДСТУ 7564:2014, яка демонструє приклад офіційної інтеграції сфери віртуальних активів у благодійність на державному рівні. Для цього спочатку було вибрано програмне середовище для реалізації застосунку, цим продуктом є Visual Studio 2019 від розробника Microsoft. Після була вибрана мова програмування C++ для реалізації блокчейну і досліджено її особливості при створенні системи. Щоб створити повноцінний блокчейн потрібна функція хешування і для програмної реалізація була обрана і досліджена функція хешування «Купина» за стандартом ДСТУ 7564:2014 та порівняна з іншими хеш-функціями. Після створення програмного коду блокчейн-системи був створений опис основних моментів і перевірена коректність роботи програми.

Отже, було досягнуто основної мети дипломної роботи – реалізовано блокчейн-систему на основі національного стандарту ДСТУ 7564:2014, щоб використати її в майбутньому для благодійної екосистеми на державному рівні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Nakamoto S. A. Peer-to-Peer Electronic Cash System. Bitcoin [Electronic resource]. – Access: https://bitcoin.org/bitcoin.pdf

2. Nathaniel Popper Digital Gold: Bitcoin and the Inside Story of the Misfits and Millionaires Trying to Reinvent Money [Electronic resource]. – Access: https://www.goodreads.com/book/show/23546676-digital-gold

3. Michael Casey, Paul Vigna The Age of Cryptocurrency: How Bitcoin and Digital Money Are Challenging the Global Economic Order [Electronic resource]. – Access: https://www.goodreads.com/book/show/22174460-the-age-of-cryptocurrency

4. Tapscott D., Tapscott A. Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World, 2016 [Electronic resource]. – Access: https://www.tandfonline.com/doi/abs/10.1080/10686967.2018.1404373

5. Усенко А. Перспективи blockchain для бізнесу та української економіки. [Електронний ресурс]. – Режим доступу: https://news.finance.ua/ua/news/-/427333/andrij-usenko-perspektyvy-blockchain-dlya-biznesu-i-ukrayinskoyi-ekonomiky

6. Peters G.W. & Panayi E. (2016). Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money. Banking Beyond Banks and Money [Electronic resource]. – Access: https://arxiv.org/pdf/1511.05740.pdf

7. Що таке блокчейн (blockchain)? [Електронний ресурс]: BTC-UP. – Режим доступу: https://btc-up.com/shho-take-blokchejn-blockchain/

8. The World Bank Group [Electronic resource]: The World Bamk.– Access: https://www.worldbank.org/

9. Blockchain Technology Defined [Electronic resource]: Built In National. – Access: https://builtin.com/blockchain

10. How secure is blockchain really? [Electronic resource]: Mike Orcuttarchive page MIT Technology Review BLOCKCHAIN. – Access: https://www.technologyreview.com/2018/04/25/143246/how-secure-is-blockchain-really/

11. Blockchain [Electronic resource]: Enigma. Paradox. Opportunity. – Access: https://www2.deloitte.com/content/dam/Deloitte/xe/Documents/technology/Blockchain.pdf

12. Біловус Л.І. Управління та організація діяльності інформаційних установ [Електронний ресурс]: Навчальний посібник. – Режим доступу^http://dspace.wunu.edu.ua/bitstream/316497/9171/1/feu_kdidu_sdid_dotudiu_NP.pdf

13. A timeline and history of blockchain technology [Electronic resource]: Tech Accelerator. – Access: https://www.techtarget.com/whatis/feature/A-timeline-and-history-of-blockchain-technology

14. Пояснення дерев та корінь Меркла [Електронний ресурс]. Binance Academy. – Режим доступу: https://academy.binance.com/uk/articles/merkle-trees-and-merkle-roots-explained

15. Alan T. Sherman, Farid Javani, Haibin Zhang, and Enis Golaszewski On the Origins and Variations of Blockchain Technologies [Electronic resource]: Cyber Defense Lab.University of Maryland, Baltimore County (UMBC). – Access: https://arxiv.org/ftp/arxiv/papers/1810/1810.06130.pdf

16. Девід Чаум - David Chaum [Електронний ресурс]. Енциклопедія – Режим доступу: https://wikiukuk.top/wiki/David_Chaum#Vault_systems

17. History of Blockchain [Electronic resource]. – Access: https://www.javatpoint.com/history-of-blockchain

18. Stuart Haber , W. Scott Stornetta How to Time-stamp a Digital Document (1991) [Electronic resource]: Journal of Cryptology. – Access: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.8740

19. Dave Bayer , W. Scott Stornetta , Stuart Haber Improving the Efficiency and Reliability of Digital Time-Stamping (1993) [Electronic resource]: Sequences II: Methods in Communication, Security and Computer Science. – Access: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.71.4891

20. Back, Adam. Hashcash - A Denial of Service Counter-Measure. [Electronic resource]. – Access: https://web.archive.org/web/20181123130639/http://www.hashcash.org/hashcash.pdf

21. History of blockchain [Electronic resource]. – Access: https://www.icaew.com/technical/technology/blockchain-and-cryptoassets/blockchain-articles/what-is-blockchain/history

22. Hal Finney Reusable Proofs of Work [Electronic resource]. – Access: https://nakamotoinstitute.org/finney/rpow/index.html

23. Narayanan, Arvind; Bonneau, Joseph; Felten, Edward; Miller, Andrew; Goldfeder, Steven (2016). Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton: Princeton University Press. [Electronic resource]. – Access: https://press.princeton.edu/books/hardcover/9780691171692/bitcoin-and-cryptocurrency-technologies

24. bitcoin mining [Electronic resource]. WhatIs.com – Access: https://www.techtarget.com/whatis/definition/Bitcoin-mining

25. БІТКОЙН (BITCOIN, BTC) [Електронний ресурс]. – Режим доступу: https://mind.ua/tags/473-bitkojn-bitcoin-btc

26. [Електронний ресурс]. – Режим доступу: https://www.moneycontrol.com/news/business/cryptocurrency/bitcoin-pizza-day-2021-some-interesting-facts-about-this-special-cryptocurrency-day-6924731.html

27. Bitcoin Pizza Day 2021: Some interesting facts about this special cryptocurrency day [Electronic resource]. – Access: https://ua.korrespondent.net/business/financial/3350239-naibilsha-birzha-bitkoiniv-MtGox-podala-zaiavu-na-likvidatsiui

28. Litecoin [Електронний ресурс]. – Режим доступу: https://coinmarketcap.com/uk/currencies/litecoin/

29. Smart contracts – [Electronic resource]. – Access: https://www.coursera.org/learn/smarter-contracts

30. Смарт-контракти: в чому родзинка? [Електронний ресурс] / Віктор Мороз // Юридична газета – Режим доступу: https://yur-gazeta.com/dumka-eksperta/cmartkontrakti-v-chomu-rodzinka.html

31. What is blockchain technology? [Electronic resource]: IBM – Access: https://www.ibm.com/topics/what-is-blockchain

32. How does blockchain work in 7 steps — A clear and simple explanation. [Electronic resource]: /Jimi S.// Good Audience – Access: https://blog.goodaudience.com/blockchain-for-beginners-what-is-blockchain-519db8c6677a

33. Що таке блокчейн? Повний посібник [Електронний ресурс]: Binance Academy – Режим доступу: https://academy.binance.com/uk/articles/what-is-blockchain-technology-a-comprehensive-guide-for-beginners#what-is-blockchain

34. Що таке ноди (вузли)? [Електронний ресурс]: EXBASE.IO – Режим доступу: https://exbase.io/uk/wiki/shho-take-nodi

35. What are Blockchain nodes? Detailed Guide [Electronic resource]: Blockchain Council – Access: https://www.blockchain-council.org/blockchain/blockchain-nodes/

36. Що таке алгоритм консенсусу на блокчейні? [Електронний ресурс]: Binance Academy – Режим доступу: https://academy.binance.com/uk/articles/what-is-a-blockchain-consensus-algorithm

37. Consensus Algorithms in Blockchain [Electronic resource]: GeeksforGeeks – Access: https://www.geeksforgeeks.org/consensus-algorithms-in-blockchain/

38. Види консенсусу Блокчейн - алгоритми [Електронний ресурс]: Guland.com.ua. – Режим доступу: https://guland.com.ua/kryptovalyuta/blockchain/vydy-konsensusu-blokcheyn.htm

39. Parma Bains Blockchain Consensus Mechanisms:A Primer for Supervisors [Electronic resource]: International Monetary Fund. – Access: https://webcache.googleusercontent.com/search?q=cache:RvHgSjA34ZsJ:https://www.elibrary.imf.org/downloadpdf/journals/063/2022/003/063.2022.issue-003en.xml+&cd=16&hl=uk&ct=clnk&gl=ua

40. Consensus Algorithms: Concept, Properties and Types [Electronic resource]: Analyticssteps. – Access: https://www.analyticssteps.com/blogs/consensus-algorithms-concept-properties-and-types

41. Що таке вилка? [Електронний ресурс]: / Ludvig// Kryptovaluta.info. – Режим доступу: https://kryptovaluta.info/uk/%D1%89%D0%BE-%D1%82%D0%B0%D0%B A%D0%B5-%D0%B2%D0%B8%D0%BB%D0%BA%D0%B0/

42. Proof-of-Stake (PoS) [Electronic resource]: Investopedia. – Access: https://www.investopedia.com/terms/p/proof-stake-pos.asp

43. Як заробити на криптовалюті: алгоритми Proof-of-Work і Proof-of-Stake [Електронний ресурс]: / Галина Чепурко// Media for creators. – Режим доступу: https://mc.today/uk/yak-zarobiti-na-kriptovalyuti-algoritmi-proof-of-work-i-proof-of-stake/

44. What Is Delegated Proof of Stake (DPoS)? [Electronic resource]: Bybit Learn. – Access: https://learn.bybit.com/blockchain/delegated-proof-of-stake-dpos/

45. Blockchain Proof Of Concept: Enterprise POC Guide [Electronic resource]: 101Blockchains. – Access: https://101blockchains.com/blockchain-proof-of-concept/

46. Proof of Capacity [Electronic resource]: GeeksforGeeks. – Access: https://www.geeksforgeeks.org/proof-of-capacity/

47. What is Proof of Authority? [Electronic resource]: Coinhouse. – Access: https://www.coinhouse.com/what-is-proof-of-authority/

48. Proof-of-Authority (PoA) [Electronic resource]: Alexandria. – Access: https://coinmarketcap.com/alexandria/glossary/proof-of-authority-poa

49. Proof-of-Authority consensus [Electronic resource]: Apla Blockchain Platform Guide. – Access: https://apla.readthedocs.io/en/latest/concepts/consensus.html

50. Proof of authority [Electronic resource]: Golden. – Access: https://golden.com/wiki/Proof_of_authority-W4BGKDV

51. What is Proof of Burn (PoB)? [Electronic resource]: DataDrivenInvestor. – Access: https://medium.datadriveninvestor.com/what-is-proof-of-burn-pob-e8f7e7dfbbfa

52. Proof of Burn Explained [Electronic resource]: Binance Academy. – Access: https://academy.binance.com/en/articles/proof-of-burn-explained

53. What Is Byzantine Fault Tolerance? [Electronic resource]: The Motley Fool. – Access: https://www.fool.com/investing/stock-market/market-sectors/financials/cryptocurrency-stocks/byzantine-fault-tolerance/

54. practical Byzantine Fault Tolerance(pBFT) [Electronic resource]: GeeksforGeeks. – Access: https://www.geeksforgeeks.org/practical-byzantine-fault-tolerancepbft/

55. Proof of Importance (PoI) [Electronic resource]: Smithandcrown. – Access: https://smithandcrown.com/glossary/proof-of-importance/

56. Proof of Importance [Electronic resource]: Moneyland. – Access: https://www.moneyland.ch/en/proof-of-importance-definition

57. Proof-of-importance (PoI) [Electronic resource]: Golden. – Access: https://golden.com/wiki/Proof-of-importance_(PoI)-639YX6M

58. Proof of Importance in Cryptocurrency [Electronic resource]: MSG. – Access: https://www.managementstudyguide.com/proof-of-importance-in-cryptocurrency.htm

59. Kathleen E. Wegrzyn Eugenia Wang. Types of Blockchain: Public, Private, or Something in Between [Electronic resource]: Manufacturing Industry Advisor Innovative Technology Insights Dashboard Insights. – Access: https://www.foley.com/en/insights/publications/2021/08/types-of-blockchain-public-private-between

60. Types of Blockchains – Decide which one is better for your Investment Needs [Electronic resource]: DataFlair. – Access: https://data-flair.training/blogs/types-of-blockchain/

61. What are the 4 different types of blockchain technology? [Electronic resource]: SearchCIO. – Access: https://www-techtarget-com.translate.goog/searchcio/feature/What-are-the-4-different-types-of-blockchain-technology?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=uk&_x_tr_pto=sc

62. What is cryptocurrency and how does it work? [Electronic resource]. – Access: https://www.kaspersky.com/resource-center/definitions/what-is-cryptocurrency

63. Огляд цифрових криптовалют [Електронний ресурс]: Bankchart.– Режим доступу: https://bankchart.com.ua/e_banking/statti/oglyad_tsifrovih_kriptovalyut#1

64. Understanding The Different Types of Cryptocurrency ? [Electronic resource]: SoFi Learn. – Access: https://www.sofi.com/learn/content/understanding-the-different-types-of-cryptocurrency/

65. What is the difference between tokens and cryptocurrencies? [Electronic resource]: CoinRivet. – Access: https://coinrivet.com/guides/guide/why-is-a-token-different-to-a-cryptocurrency/

66. Digital Assets: Cryptocurrencies vs. Tokens [Electronic resource]: Cryptopedia. – Access: https://www.gemini.com/cryptopedia/cryptocurrencies-vs-tokens-difference

67. What is cryptocurrency trading and how does it work? [Electronic resource]: Bitcoin. – Access: https://www.ig.com/en/cryptocurrency-trading/what-is-cryptocurrency-trading-how-does-it-work

68. Frequently Asked Questions [Electronic resource]: IG. – Access: https://bitcoin.org/en/faq#general

69. Вадим Попов. Що таке криптовалюта? [Електронний ресурс]. РадіоСвобода – Режим доступу: https://www.radiosvoboda.org/a/details/28742278.html

70. 8 Pros and Cons of Bitcoin [Electronic resource]: MintLife. – Access: https://mint.intuit.com/blog/investments/pros-and-cons-of-bitcoin/

71. Litecoin (LTC) [Electronic resource]: Investopedia. – Access: https://www.investopedia.com/articles/investing/040515/what-litecoin-and-how-does-it-work.asp

72. What is Litecoin? [Electronic resource]: CoinHouse. – Access: https://www.coinhouse.com/litecoin/

73. Litecoin [Electronic resource]: Criptoeq. – Access: https://www.cryptoeq.io/corereports/litecoin-abridged

74. Pros and Cons of Investing in Litecon [Electronic resource]: Trading Education. – Access: https://trading-education.com/pros-and-cons-of-investing-in-litecoin-will-it-be-a-millionaire-maker#h_60118580320148162868977173 6

75. 5 Ethereum Features [Electronic resource]: DataDrivenInvestor. – Access: https://medium.datadriveninvestor.com/5-ethereum-features-76da9462b319

76. What is Ethereum: Understanding Its Features and Applications [Electronic resource]: SimplLearn. – Access: https://www.simplilearn.com/tutorials/blockchain-tutorial/what-is-ethereum

77. Our World Is Changing. Together, We Can Change It For The Better. [Electronic resource]: Cardano. – Access: https://cardano.org/

78. What is Cardano? [Electronic resource]: STILT. – Access: https://www.stilt.com/blog/2021/10/what-is-cardano/

79. What is a stablecoin? [Electronic resource]: CoinBase. – Access: https://www.coinbase.com/learn/crypto-basics/what-is-a-stablecoin

80. Що таке стейблкоїни і навіщо їх використовують при торгівлі [Електронний ресурс]: Fibi – Режим доступу: https://fibi.tech/news/kriptovalyuti/sho-take-stejblkoini-i-navisho-yih-vikoristovuyut-pri-torgivli

81. Stablecoin [Electronic resource]: / Adam Hayes // Investopedia. – Access: https://www.investopedia.com/terms/s/stablecoin.asp

82. Що таке стейблкоїн? Короткий огляд спеціально для вас [Електронний ресурс]: Gate.io – Режим доступу: https://www.gate.io/uk/blog_detail/426/What-Is-a-Stablecoin--A-short-overview-just-for-you

83. Ultimate Cheat Sheet To Start Coding In Blockchain: How Can We C? [Electronic resource]. – Access: https://medium.com/neptune-insights/ultimate-cheat-sheet-to-start-coding-in-blockchain-how-can-we-c-9b4eeff723f8

84. Building a Blockchain? Consider C++ [Electronic resource]: Blockchain Works. – Access: https://blockchain.works-hub.com/learn/Building-a-Blockchain-Consider-C-

85. Introduction to Visual Studio [Electronic resource]: GeeksforGeeks. – Access: https://www.geeksforgeeks.org/introduction-to-visual-studio/

86. Blockchain Hash Function [Electronic resource]: JavaTpoint. – Access: https://www.javatpoint.com/blockchain-hash-function

87. ДСТУ 7564:2014. Інформаційні технології. Криптографічний захист інформації. Функція гешування. — Введ. 01–04–2015. — К. : Мінекономрозвитку України, 2015.

ДОДАДОК А
ПРОГРАМНИЙ КОД РЕАЛІЗАЦІЇ ТЕХНОЛОГІЇ БЛОКЧЕЙН

```
// Файл Block.h
#include <cstdint>
#include <iostream>
using namespace std;
class CBlock final {
public:
	string PrevHash;
	CBlock(std::uint32_t indexIn, const string & dataIn);
	void SetAsGenesis();
	uint32_t GetIndex() const;
	time_t GetTime() const;
	string GetHash() const;
	void DOMine(uint32_t difficulty);
private:
	uint32_t m_index;
	int64_t m_nonce;
	string m_data;
	string m_hash;
	time_t m_time;
	const string CalculateHash() const;};

// Файл BlockChain.h
#include <cstdint>
#include <vector>
#include "Block.h"
using namespace std;
```

```
class CBlockChain {
public:
	CBlockChain();
	void AddGenesis();
	void AddBlock(CBlock newBlock);
private:
	uint32_t m_difficulty;
	vector<CBlock> m_chain;
	const CBlock GetLastBlock() const;
	const uint32_t GetChainSize() const;};

// Файл Crypto.h
#pragma once
#include <cstdint>
#include <array>
using namespace std;
namespace Crypt{
	using Hash = array<uint8_t, 32>;
	using PublicKey = array<uint8_t, 25>;
	using SecretKey = array<uint8_t, 32>;
	using Signature = array<uint8_t, 65>;}

// Файл Kupyna.h
#ifndef SRC_KUPYNA_H_
#define SRC_KUPYNA_H_
#include <stdlib.h>
#include <limits.h>
#define ROWS 8
#define NB_512 8  ///< Number of 8-byte words in state for <=256-bit hash code.
#define NB_1024 16  ///< Number of 8-byte words in state for <=512-bit hash code.
```

```
#define STATE_BYTE_SIZE_512 (ROWS * NB_512)
#define STATE_BYTE_SIZE_1024 (ROWS * NB_1024)
#define NR_512 10  ///< Number of rounds for 512-bit state.
#define NR_1024 14  ///< Number of rounds for 1024-bit state.
#define REDUCTION_POLYNOMIAL 0x011d  /* x^8 + x^4 + x^3 + x^2 + 1 */
#if (ULLONG_MAX != 0xFFFFFFFFFFFFFFFFULL)
#error "Architecture not supported. Required type to fit 64 bits."
#endif
#define BITS_IN_WORD 64
#if (CHAR_BIT != 8)
#error "Architecture not supported. Required type to fit 8 bits."
#endif
#define BITS_IN_BYTE 8
typedef unsigned char uint8_t;
typedef unsigned long long uint64_t;
typedef struct {
    uint8_t state[NB_1024][ROWS];  ///< Hash function internal state (of maximum
possible size to fit for all modes of operation).
    size_t nbytes;  ///< Number of bytes currently located in state.
    size_t data_nbytes;  ///< Number of bytes in input data sequence.
    uint8_t padding[STATE_BYTE_SIZE_1024 * 2];  ///< Space for extra bytes and
padding.
    size_t pad_nbytes;  ///< Number of bytes currently located in padding buffer.
    size_t hash_nbits;  ///< Hash code bit length.
    int columns;  ///< Number of columns (8-byte vectors) located in internal state.
    int rounds;  ///< Number of rounds for current mode of operation.
} kupyna_t;
int KupynaInit(size_t hash_nbits, kupyna_t* ctx);
void KupynaHash(kupyna_t* ctx, uint8_t* data, size_t msg_nbits, uint8_t* hash_code);
#endif  /* SRC_KUPYNA_H_ */
```

```
// Файл Tables.h
#pragma once
#define KUPYNA_TABLES_H_
#include "kupyna.h"
extern uint8_t mds_matrix[8][8];
extern uint8_t sboxes[4][256];

// Файл Transication.h
#include <cstdint>
#include <vector>
#include "Block.h"
#include "Crypto.h"
using namespace std;
class CTXBase {
public:
	virtual void SetNull();
	virtual bool IsNull() const;
	virtual string ToString() const = 0;};
class CTXInput final : CTXBase {
private:
	uint32_t m_index;
	uint64_t m_amount;
	vector<uint8_t> m_datas;
public:
	CTXInput() {
		this->SetNull();}
	CTXInput(uint32_t index, uint64_t amount, vector<uint8_t> datas) {
		this->m_index = index;
		this->m_amount = amount;
```

```
            this->m_datas = datas;}
        bool operator==(const CTXInput& other) const;
        void SetNull();
        bool IsNull() const;
        uint32_t GetIndex() const;
        uint64_t GetAmount() const;
        vector<uint8_t> GetDatas() const;
        string ToString() const;};
class CTXOutput final : CTXBase {
private:
        Crypt::PublicKey m_key;
        uint64_t m_amount;
public:
        CTXOutput() {
            this->SetNull();}
        CTXOutput(Crypt::PublicKey key, uint64_t amount) {
            this->m_key = key;
            this->m_amount = amount;}
        bool operator==(const CTXOutput& other) const;
        void SetNull();
        bool IsNull() const;
        Crypt::PublicKey GetKey() const;
        uint64_t GetAmount() const;
        string ToString() const;};
class CTXPrefix {
protected:
        uint64_t m_time;
        vector<CTXInput> m_inputs;
        vector<CTXOutput> m_outputs;
        vector<uint8_t> m_extras;
```

```
public:
        uint64_t GetTime() const;
        vector<CTXInput> GetInputs() const;
        vector<CTXOutput> GetOutputs() const;
        vector<uint8_t> GetExtras() const;};
class CTransactionX final : public CTXPrefix {
private:
        int m_version;
        int m_lockTime;
        Crypt::PublicKey m_from;
        Crypt::PublicKey m_to;
public:
        CTransactionX() {
                this->SetNull();}
        bool operator==(const CTransactionX& other) const;
        int GetVersion() const;
        int GetLockTime() const;
        void SetNull();
        bool IsNull() const;
        string ToString() const;};

// Файл Block.cpp
#include <stdio.h>
#include <memory.h>
#include <ctime>
#include <sstream>
#include "Block.h"
#include "Kupyna.h"
#include <iostream>
```

```
#include <string>CBlock::CBlock(uint32_t indexIn, const string & dataIn) :
m_index(indexIn), m_data(dataIn) {
	this->m_nonce = -1;
	this->m_time = time(nullptr);}
uint32_t CBlock::GetIndex() const {
	return this->m_index;}
time_t CBlock::GetTime() const {
	return this->m_time;}
string CBlock::GetHash() const {
	return this->m_hash;}
void CBlock::SetAsGenesis() {
	this->m_nonce = -1;
	this->m_time = time(nullptr);
	this->m_index = 0;
	this->m_data = "";
	this->m_hash = "Genesis";}
void CBlock::DOMine(uint32_t difficulty) {
	char * cstr = new char[difficulty + 1];
	for (uint32_t i = 0; i < difficulty; ++i) {
		cstr[i] = '0';}
	cstr[difficulty] = '\0';
	string str(cstr);
	do {
		this->m_nonce++;
		this->m_hash = CalculateHash();
	} while (!str.compare(this->m_hash.substr(0, difficulty)));
	cout << "Block mined: " << this->m_hash << endl;}
inline const string CBlock::CalculateHash() const {
	stringstream ss;
	string s1;
```

```
	kupyna_t ctx;
	uint8_t hash_code[512 / 8];
	ss << this->m_index << this->m_time << this->m_data << this->m_nonce << this->PrevHash;
	KupynaInit(200, &ctx);
	uint8_t test[256];
	s1 = ss.str();
	std::memcpy(test, s1.data(), s1.size());
	KupynaHash(&ctx, test, 512, hash_code);
	string sq((char*)(256,hash_code));
	return sq;}
void print(int data_len, uint8_t data[]){
	int i = 0;
	int data_size = data_len / BITS_IN_BYTE;
	for (i = 0; i < data_size; i++){
		if (!(i % 16)) printf("    ");
		printf("%02X", (unsigned int)data[i]);
		if (!((i + 1) % 16)) printf("\n");};
	if (data_len % BITS_IN_BYTE != 0){
		if (!(i % 16)) printf("    ");
printf("%02X", (unsigned int)((data[i]) & (~((1 << (BITS_IN_BYTE - (data_len % BITS_IN_BYTE))) - 1))));
		if (!((i + 1) % 16)) printf("\n");};
	printf("\n");};

// Файл BlockChain.cpp
#include "BlockChain.h"
#include <string>
CBlockChain::CBlockChain() {
	//this->m_chain.emplace_back(CBlock(0, "Genesis Block"));
```

```
        this->m_difficulty = 10;}
void CBlockChain::AddGenesis() {
        CBlock genesis(0, "");
        genesis.SetAsGenesis();
        this->m_chain.emplace_back(genesis);}
void CBlockChain::AddBlock(CBlock newBlock) {
        newBlock.PrevHash = GetLastBlock().GetHash();
        newBlock.DOMine(this->m_difficulty);
        this->m_chain.push_back(newBlock);}
const CBlock CBlockChain::GetLastBlock() const {
        return this->m_chain.back();}
const uint32_t CBlockChain::GetChainSize() const {
        return this->m_chain.size();}

// Файл Kupyna.cpp
#include <memory.h>
#include <stdio.h>
#include "Kupyna.h"
#include "Tables.h"
int KupynaInit(size_t hash_nbits, kupyna_t* ctx) {
   if ((hash_nbits % 8 != 0) || (hash_nbits > 512)) {
      return -1;}
   if (hash_nbits <= 256) {
      ctx->rounds = NR_512;
      ctx->columns = NB_512;
      ctx->nbytes = STATE_BYTE_SIZE_512; }
   else {
      ctx->rounds = NR_1024;
      ctx->columns = NB_1024;
      ctx->nbytes = STATE_BYTE_SIZE_1024;   }
```

```
    ctx->hash_nbits = hash_nbits;
    memset(ctx->state, 0, ctx->nbytes);
    ctx->state[0][0] = ctx->nbytes;
    return 0;}
void SubBytes(uint8_t state[NB_1024][ROWS], int columns) {
    int i, j;
    uint8_t temp[NB_1024];
    for (i = 0; i < ROWS; ++i) {
        for (j = 0; j < columns; ++j) {
            state[j][i] = sboxes[i % 4][state[j][i]]; }  }}
void ShiftBytes(uint8_t state[NB_1024][ROWS], int columns) {
    int i, j;
    uint8_t temp[NB_1024];
    int shift = -1;
    for (i = 0; i < ROWS; ++i) {
        if ((i == ROWS - 1) && (columns == NB_1024)) {
            shift = 11; }
        else {
            ++shift;}
        for (j = 0; j < columns; ++j) {
            temp[(j + shift) % columns] = state[j][i]; }
        for (j = 0; j < columns; ++j) {
            state[j][i] = temp[j];   }}}
uint8_t MultiplyGF(uint8_t x, uint8_t y) {
    int i;
    uint8_t r = 0;
    uint8_t hbit = 0;
    for (i = 0; i < BITS_IN_BYTE; ++i) {
        if ((y & 0x1) == 1)
            r ^= x;
```

```
        hbit = x & 0x80;
        x <<= 1;
        if (hbit == 0x80)
            x ^= REDUCTION_POLYNOMIAL;
        y >>= 1; }
    return r;}
void MixColumns(uint8_t state[NB_1024][ROWS], int columns) {
    int i, row, col, b;
    uint8_t product;
    uint8_t result[ROWS];
    for (col = 0; col < columns; ++col) {
        memset(result, ROWS, 0);
        for (row = ROWS - 1; row >= 0; --row) {
            product = 0;
            for (b = ROWS - 1; b >= 0; --b) {
                product ^= MultiplyGF(state[col][b], mds_matrix[row][b]); }
            result[row] = product;}
        for (i = 0; i < ROWS; ++i) {
            state[col][i] = result[i];  }  }}
void AddRoundConstantP(uint8_t state[NB_1024][ROWS], int columns, int round){
    int i;
    for (i = 0; i < columns; ++i) {
        state[i][0] ^= (i * 0x10) ^ round;  }}
void AddRoundConstantQ(uint8_t state[NB_1024][ROWS], int columns, int round){
    int j;
    uint64_t* s = (uint64_t*)state;
    for (j = 0; j < columns; ++j) {
        s[j] = s[j] + (0x00F0F0F0F0F0F0F3ULL ^
            ((((columns - j - 1) * 0x10ULL) ^ round) << (7 * 8)));}}
void P(kupyna_t* ctx, uint8_t state[NB_1024][ROWS]) {
```

```
    int i;
    for (i = 0; i < ctx->rounds; ++i) {
        AddRoundConstantP(state, ctx->columns, i);
        SubBytes(state, ctx->columns);
        ShiftBytes(state, ctx->columns);
        MixColumns(state, ctx->columns);}}
void Q(kupyna_t* ctx, uint8_t state[NB_1024][ROWS]) {
    int i;
    for (i = 0; i < ctx->rounds; ++i) {
        AddRoundConstantQ(state, ctx->columns, i);
        SubBytes(state, ctx->columns);
        ShiftBytes(state, ctx->columns);
        MixColumns(state, ctx->columns);}}
int Pad(kupyna_t* ctx, uint8_t* data, size_t msg_nbits) {
    int i;
    int mask;
    int pad_bit;
    int extra_bits;
    int zero_nbytes;
    size_t msg_nbytes = msg_nbits / BITS_IN_BYTE;
    size_t nblocks = msg_nbytes / ctx->nbytes;
    ctx->pad_nbytes = msg_nbytes - (nblocks * ctx->nbytes);
    ctx->data_nbytes = msg_nbytes - ctx->pad_nbytes;
    uint8_t* pad_start = data + ctx->data_nbytes;
    extra_bits = msg_nbits % BITS_IN_BYTE;
    if (extra_bits) {
        ctx->pad_nbytes += 1;}
    memcpy(ctx->padding, pad_start, ctx->pad_nbytes);
    extra_bits = msg_nbits % BITS_IN_BYTE;
    if (extra_bits) {
```

```
        mask = ~(0xFF >> (extra_bits));
        pad_bit = 1 << (7 - extra_bits);
        ctx->padding[ctx->pad_nbytes - 1] = (ctx->padding[ctx->pad_nbytes - 1] & mask) |
pad_bit;}
    else {
        ctx->padding[ctx->pad_nbytes] = 0x80;
        ctx->pad_nbytes += 1;  }
    msg_nbits = (-1) * msg_nbits;
    zero_nbytes = ((msg_nbits - 97) % (ctx->nbytes * BITS_IN_BYTE)) /
BITS_IN_BYTE;
    memset(ctx->padding + ctx->pad_nbytes, 0, zero_nbytes);
    ctx->pad_nbytes += zero_nbytes;
    for (i = 0; i < (96 / 8); ++i, ++ctx->pad_nbytes) {
        if (i < sizeof(size_t)) {
            ctx->padding[ctx->pad_nbytes] = (msg_nbits >> (i * 8)) & 0xFF; }
        else {
            ctx->padding[ctx->pad_nbytes] = 0;   }}
    return 0;}


void Digest(kupyna_t* ctx, uint8_t* data) {
    int b, i, j;
    uint8_t temp1[NB_1024][ROWS];
    uint8_t temp2[NB_1024][ROWS];
    for (b = 0; b < ctx->data_nbytes; b += ctx->nbytes) {
        for (i = 0; i < ROWS; ++i) {
            for (j = 0; j < ctx->columns; ++j) {
                temp1[j][i] = ctx->state[j][i] ^ data[b + j * ROWS + i];
                temp2[j][i] = data[b + j * ROWS + i];   }  }
        P(ctx, temp1);
```

```
        Q(ctx, temp2);
        for (i = 0; i < ROWS; ++i) {
            for (j = 0; j < ctx->columns; ++j) {
                ctx->state[j][i] ^= temp1[j][i] ^ temp2[j][i];    }  }   }
    /* Process extra bytes in padding. */
    for (b = 0; b < ctx->pad_nbytes; b += ctx->nbytes) {
        for (i = 0; i < ROWS; ++i) {
            for (j = 0; j < ctx->columns; ++j) {
                temp1[j][i] = ctx->state[j][i] ^ ctx->padding[b + j * ROWS + i];
                temp2[j][i] = ctx->padding[b + j * ROWS + i]; }   }
        P(ctx, temp1);
        Q(ctx, temp2);
        for (i = 0; i < ROWS; ++i) {
            for (j = 0; j < ctx->columns; ++j) {
                ctx->state[j][i] ^= temp1[j][i] ^ temp2[j][i];}  }    } }
void Trunc(kupyna_t* ctx, uint8_t* hash_code) {
    int i;
    size_t hash_nbytes = ctx->hash_nbits / BITS_IN_BYTE;
    memcpy(hash_code, (uint8_t*)ctx->state + ctx->nbytes - hash_nbytes, hash_nbytes);}
void OutputTransformation(kupyna_t* ctx, uint8_t* hash_code) {
    int i, j;
    uint8_t temp[NB_1024][ROWS];
    memcpy(temp, ctx->state, ROWS * NB_1024);
    P(ctx, temp);
    for (i = 0; i < ROWS; ++i) {
        for (j = 0; j < ctx->columns; ++j) {
            ctx->state[j][i] ^= temp[j][i];  }  }
    Trunc(ctx, hash_code);}
void KupynaHash(kupyna_t* ctx, uint8_t* data, size_t msg_bit_len, uint8_t* hash_code) {
    /* Reinitialize internal state. */
```

```
    memset(ctx->state, 0, ctx->nbytes);
    ctx->state[0][0] = ctx->nbytes;
    Pad(ctx, data, msg_bit_len);
    Digest(ctx, data);
    OutputTransformation(ctx, hash_code);}

// Файл Tables.cpp
#include "kupyna.h"
uint8_t mds_matrix[8][8] = {
        {0x01, 0x01, 0x05, 0x01, 0x08, 0x06, 0x07, 0x04},
        {0x04, 0x01, 0x01, 0x05, 0x01, 0x08, 0x06, 0x07},
        {0x07, 0x04, 0x01, 0x01, 0x05, 0x01, 0x08, 0x06},
        {0x06, 0x07, 0x04, 0x01, 0x01, 0x05, 0x01, 0x08},
        {0x08, 0x06, 0x07, 0x04, 0x01, 0x01, 0x05, 0x01},
        {0x01, 0x08, 0x06, 0x07, 0x04, 0x01, 0x01, 0x05},
        {0x05, 0x01, 0x08, 0x06, 0x07, 0x04, 0x01, 0x01},
        {0x01, 0x05, 0x01, 0x08, 0x06, 0x07, 0x04, 0x01}};
uint8_t sboxes[4][256] = { {
        0xa8, 0x43, 0x5f, 0x06, 0x6b, 0x75, 0x6c, 0x59, 0x71, 0xdf, 0x87, 0x95, 0x17,
0xf0, 0xd8, 0x09,
        0x6d, 0xf3, 0x1d, 0xcb, 0xc9, 0x4d, 0x2c, 0xaf, 0x79, 0xe0, 0x97, 0xfd, 0x6f, 0x4b,
0x45, 0x39,
        0x3e, 0xdd, 0xa3, 0x4f, 0xb4, 0xb6, 0x9a, 0x0e, 0x1f, 0xbf, 0x15, 0xe1, 0x49,
0xd2, 0x93, 0xc6,
        0x92, 0x72, 0x9e, 0x61, 0xd1, 0x63, 0xfa, 0xee, 0xf4, 0x19, 0xd5, 0xad, 0x58,
0xa4, 0xbb, 0xa1,
        0xdc, 0xf2, 0x83, 0x37, 0x42, 0xe4, 0x7a, 0x32, 0x9c, 0xcc, 0xab, 0x4a, 0x8f, 0x6e,
0x04, 0x27,
        0x2e, 0xe7, 0xe2, 0x5a, 0x96, 0x16, 0x23, 0x2b, 0xc2, 0x65, 0x66, 0x0f, 0xbc,
0xa9, 0x47, 0x41,
```

0x34, 0x48, 0xfc, 0xb7, 0x6a, 0x88, 0xa5, 0x53, 0x86, 0xf9, 0x5b, 0xdb, 0x38, 0x7b, 0xc3, 0x1e,

0x22, 0x33, 0x24, 0x28, 0x36, 0xc7, 0xb2, 0x3b, 0x8e, 0x77, 0xba, 0xf5, 0x14, 0x9f, 0x08, 0x55,

0x9b, 0x4c, 0xfe, 0x60, 0x5c, 0xda, 0x18, 0x46, 0xcd, 0x7d, 0x21, 0xb0, 0x3f, 0x1b, 0x89, 0xff,

0xeb, 0x84, 0x69, 0x3a, 0x9d, 0xd7, 0xd3, 0x70, 0x67, 0x40, 0xb5, 0xde, 0x5d, 0x30, 0x91, 0xb1,

0x78, 0x11, 0x01, 0xe5, 0x00, 0x68, 0x98, 0xa0, 0xc5, 0x02, 0xa6, 0x74, 0x2d, 0x0b, 0xa2, 0x76,

0xb3, 0xbe, 0xce, 0xbd, 0xae, 0xe9, 0x8a, 0x31, 0x1c, 0xec, 0xf1, 0x99, 0x94, 0xaa, 0xf6, 0x26,

0x2f, 0xef, 0xe8, 0x8c, 0x35, 0x03, 0xd4, 0x7f, 0xfb, 0x05, 0xc1, 0x5e, 0x90, 0x20, 0x3d, 0x82,

0xf7, 0xea, 0x0a, 0x0d, 0x7e, 0xf8, 0x50, 0x1a, 0xc4, 0x07, 0x57, 0xb8, 0x3c, 0x62, 0xe3, 0xc8,

0xac, 0x52, 0x64, 0x10, 0xd0, 0xd9, 0x13, 0x0c, 0x12, 0x29, 0x51, 0xb9, 0xcf, 0xd6, 0x73, 0x8d,

0x81, 0x54, 0xc0, 0xed, 0x4e, 0x44, 0xa7, 0x2a, 0x85, 0x25, 0xe6, 0xca, 0x7c, 0x8b, 0x56, 0x80},{

0xce, 0xbb, 0xeb, 0x92, 0xea, 0xcb, 0x13, 0xc1, 0xe9, 0x3a, 0xd6, 0xb2, 0xd2, 0x90, 0x17, 0xf8,

0x42, 0x15, 0x56, 0xb4, 0x65, 0x1c, 0x88, 0x43, 0xc5, 0x5c, 0x36, 0xba, 0xf5, 0x57, 0x67, 0x8d,

0x31, 0xf6, 0x64, 0x58, 0x9e, 0xf4, 0x22, 0xaa, 0x75, 0x0f, 0x02, 0xb1, 0xdf, 0x6d, 0x73, 0x4d,

0x7c, 0x26, 0x2e, 0xf7, 0x08, 0x5d, 0x44, 0x3e, 0x9f, 0x14, 0xc8, 0xae, 0x54, 0x10, 0xd8, 0xbc,

0x1a, 0x6b, 0x69, 0xf3, 0xbd, 0x33, 0xab, 0xfa, 0xd1, 0x9b, 0x68, 0x4e, 0x16, 0x95, 0x91, 0xee,

0x4c, 0x63, 0x8e, 0x5b, 0xcc, 0x3c, 0x19, 0xa1, 0x81, 0x49, 0x7b, 0xd9, 0x6f, 0x37, 0x60, 0xca,

0xe7, 0x2b, 0x48, 0xfd, 0x96, 0x45, 0xfc, 0x41, 0x12, 0x0d, 0x79, 0xe5, 0x89, 0x8c, 0xe3, 0x20,

0x30, 0xdc, 0xb7, 0x6c, 0x4a, 0xb5, 0x3f, 0x97, 0xd4, 0x62, 0x2d, 0x06, 0xa4, 0xa5, 0x83, 0x5f,

0x2a, 0xda, 0xc9, 0x00, 0x7e, 0xa2, 0x55, 0xbf, 0x11, 0xd5, 0x9c, 0xcf, 0x0e, 0x0a, 0x3d, 0x51,

0x7d, 0x93, 0x1b, 0xfe, 0xc4, 0x47, 0x09, 0x86, 0x0b, 0x8f, 0x9d, 0x6a, 0x07, 0xb9, 0xb0, 0x98,

0x18, 0x32, 0x71, 0x4b, 0xef, 0x3b, 0x70, 0xa0, 0xe4, 0x40, 0xff, 0xc3, 0xa9, 0xe6, 0x78, 0xf9,

0x8b, 0x46, 0x80, 0x1e, 0x38, 0xe1, 0xb8, 0xa8, 0xe0, 0x0c, 0x23, 0x76, 0x1d, 0x25, 0x24, 0x05,

0xf1, 0x6e, 0x94, 0x28, 0x9a, 0x84, 0xe8, 0xa3, 0x4f, 0x77, 0xd3, 0x85, 0xe2, 0x52, 0xf2, 0x82,

0x50, 0x7a, 0x2f, 0x74, 0x53, 0xb3, 0x61, 0xaf, 0x39, 0x35, 0xde, 0xcd, 0x1f, 0x99, 0xac, 0xad,

0x72, 0x2c, 0xdd, 0xd0, 0x87, 0xbe, 0x5e, 0xa6, 0xec, 0x04, 0xc6, 0x03, 0x34, 0xfb, 0xdb, 0x59,

0xb6, 0xc2, 0x01, 0xf0, 0x5a, 0xed, 0xa7, 0x66, 0x21, 0x7f, 0x8a, 0x27, 0xc7, 0xc0, 0x29, 0xd7},{

0x93, 0xd9, 0x9a, 0xb5, 0x98, 0x22, 0x45, 0xfc, 0xba, 0x6a, 0xdf, 0x02, 0x9f, 0xdc, 0x51, 0x59,

0x4a, 0x17, 0x2b, 0xc2, 0x94, 0xf4, 0xbb, 0xa3, 0x62, 0xe4, 0x71, 0xd4, 0xcd, 0x70, 0x16, 0xe1,

0x49, 0x3c, 0xc0, 0xd8, 0x5c, 0x9b, 0xad, 0x85, 0x53, 0xa1, 0x7a, 0xc8, 0x2d, 0xe0, 0xd1, 0x72,

0xa6, 0x2c, 0xc4, 0xe3, 0x76, 0x78, 0xb7, 0xb4, 0x09, 0x3b, 0x0e, 0x41, 0x4c, 0xde, 0xb2, 0x90,

0x25, 0xa5, 0xd7, 0x03, 0x11, 0x00, 0xc3, 0x2e, 0x92, 0xef, 0x4e, 0x12, 0x9d, 0x7d, 0xcb, 0x35,

0x10, 0xd5, 0x4f, 0x9e, 0x4d, 0xa9, 0x55, 0xc6, 0xd0, 0x7b, 0x18, 0x97, 0xd3, 0x36, 0xe6, 0x48,

0x56, 0x81, 0x8f, 0x77, 0xcc, 0x9c, 0xb9, 0xe2, 0xac, 0xb8, 0x2f, 0x15, 0xa4, 0x7c, 0xda, 0x38,

0x1e, 0x0b, 0x05, 0xd6, 0x14, 0x6e, 0x6c, 0x7e, 0x66, 0xfd, 0xb1, 0xe5, 0x60, 0xaf, 0x5e, 0x33,

0x87, 0xc9, 0xf0, 0x5d, 0x6d, 0x3f, 0x88, 0x8d, 0xc7, 0xf7, 0x1d, 0xe9, 0xec, 0xed, 0x80, 0x29,

0x27, 0xcf, 0x99, 0xa8, 0x50, 0x0f, 0x37, 0x24, 0x28, 0x30, 0x95, 0xd2, 0x3e, 0x5b, 0x40, 0x83,

0xb3, 0x69, 0x57, 0x1f, 0x07, 0x1c, 0x8a, 0xbc, 0x20, 0xeb, 0xce, 0x8e, 0xab, 0xee, 0x31, 0xa2,

0x73, 0xf9, 0xca, 0x3a, 0x1a, 0xfb, 0x0d, 0xc1, 0xfe, 0xfa, 0xf2, 0x6f, 0xbd, 0x96, 0xdd, 0x43,

0x52, 0xb6, 0x08, 0xf3, 0xae, 0xbe, 0x19, 0x89, 0x32, 0x26, 0xb0, 0xea, 0x4b, 0x64, 0x84, 0x82,

0x6b, 0xf5, 0x79, 0xbf, 0x01, 0x5f, 0x75, 0x63, 0x1b, 0x23, 0x3d, 0x68, 0x2a, 0x65, 0xe8, 0x91,

0xf6, 0xff, 0x13, 0x58, 0xf1, 0x47, 0x0a, 0x7f, 0xc5, 0xa7, 0xe7, 0x61, 0x5a, 0x06, 0x46, 0x44,

0x42, 0x04, 0xa0, 0xdb, 0x39, 0x86, 0x54, 0xaa, 0x8c, 0x34, 0x21, 0x8b, 0xf8, 0x0c, 0x74, 0x67},{

0x68, 0x8d, 0xca, 0x4d, 0x73, 0x4b, 0x4e, 0x2a, 0xd4, 0x52, 0x26, 0xb3, 0x54, 0x1e, 0x19, 0x1f,

0x22, 0x03, 0x46, 0x3d, 0x2d, 0x4a, 0x53, 0x83, 0x13, 0x8a, 0xb7, 0xd5, 0x25, 0x79, 0xf5, 0xbd,

0x58, 0x2f, 0x0d, 0x02, 0xed, 0x51, 0x9e, 0x11, 0xf2, 0x3e, 0x55, 0x5e, 0xd1, 0x16, 0x3c, 0x66,

```
        0x70, 0x5d, 0xf3, 0x45, 0x40, 0xcc, 0xe8, 0x94, 0x56, 0x08, 0xce, 0x1a, 0x3a,
0xd2, 0xe1, 0xdf,
        0xb5, 0x38, 0x6e, 0x0e, 0xe5, 0xf4, 0xf9, 0x86, 0xe9, 0x4f, 0xd6, 0x85, 0x23, 0xcf,
0x32, 0x99,
        0x31, 0x14, 0xae, 0xee, 0xc8, 0x48, 0xd3, 0x30, 0xa1, 0x92, 0x41, 0xb1, 0x18,
0xc4, 0x2c, 0x71,
        0x72, 0x44, 0x15, 0xfd, 0x37, 0xbe, 0x5f, 0xaa, 0x9b, 0x88, 0xd8, 0xab, 0x89,
0x9c, 0xfa, 0x60,
        0xea, 0xbc, 0x62, 0x0c, 0x24, 0xa6, 0xa8, 0xec, 0x67, 0x20, 0xdb, 0x7c, 0x28,
0xdd, 0xac, 0x5b,
        0x34, 0x7e, 0x10, 0xf1, 0x7b, 0x8f, 0x63, 0xa0, 0x05, 0x9a, 0x43, 0x77, 0x21,
0xbf, 0x27, 0x09,
        0xc3, 0x9f, 0xb6, 0xd7, 0x29, 0xc2, 0xeb, 0xc0, 0xa4, 0x8b, 0x8c, 0x1d, 0xfb, 0xff,
0xc1, 0xb2,
        0x97, 0x2e, 0xf8, 0x65, 0xf6, 0x75, 0x07, 0x04, 0x49, 0x33, 0xe4, 0xd9, 0xb9,
0xd0, 0x42, 0xc7,
        0x6c, 0x90, 0x00, 0x8e, 0x6f, 0x50, 0x01, 0xc5, 0xda, 0x47, 0x3f, 0xcd, 0x69,
0xa2, 0xe2, 0x7a,
        0xa7, 0xc6, 0x93, 0x0f, 0x0a, 0x06, 0xe6, 0x2b, 0x96, 0xa3, 0x1c, 0xaf, 0x6a,
0x12, 0x84, 0x39,
        0xe7, 0xb0, 0x82, 0xf7, 0xfe, 0x9d, 0x87, 0x5c, 0x81, 0x35, 0xde, 0xb4, 0xa5, 0xfc,
0x80, 0xef,
        0xcb, 0xbb, 0x6b, 0x76, 0xba, 0x5a, 0x7d, 0x78, 0x0b, 0x95, 0xe3, 0xad, 0x74,
0x98, 0x3b, 0x36,
        0x64, 0x6d, 0xdc, 0xf0, 0x59, 0xa9, 0x4c, 0x17, 0x7f, 0x91, 0xb8, 0xc9, 0x57,
0x1b, 0xe0, 0x61}};

// Файл Transaction.cpp
#include "Transaction.h"
void CTXBase::SetNull(){}
```

```
bool CTXBase::IsNull() const{
	return false;}
string CTXBase::ToString() const {
	return "CTXBase";}
bool CTXInput::operator==(const CTXInput & other) const {
	return (this->m_index == other.GetIndex() && this->m_amount ==
other.GetAmount() && this->m_datas == other.GetDatas());}
void CTXInput::SetNull() {
	this->m_index = -1;
	this->m_amount = 0;}
bool CTXInput::IsNull() const {
	return this->m_index == -1;}
uint32_t CTXInput::GetIndex() const {
	return this->m_index;}
uint64_t CTXInput::GetAmount() const {
	return this->m_amount;}
vector<uint8_t> CTXInput::GetDatas() const {
	return this->m_datas;}
string CTXInput::ToString() const {
	return string();}
bool CTXOutput::operator==(const CTXOutput & other) const {
	return (this->m_key == other.GetKey() && this->m_amount ==
other.GetAmount());}
void CTXOutput::SetNull() {
	this->m_amount = 0;}
bool CTXOutput::IsNull() const {
	return false;}
Crypt::PublicKey CTXOutput::GetKey() const {
	return this->m_key;}
uint64_t CTXOutput::GetAmount() const {
```

```
        return this->m_amount;}
string CTXOutput::ToString() const {
        return string();}
uint64_t CTXPrefix::GetTime() const {
        return this->m_time;}
vector<CTXInput> CTXPrefix::GetInputs() const {
        return this->m_inputs;}
vector<CTXOutput> CTXPrefix::GetOutputs() const {
        return this->m_outputs;}
vector<uint8_t> CTXPrefix::GetExtras() const {
        return this->m_extras;}
bool CTransactionX::operator==(const CTransactionX & other) const {
        return (this->m_time == other.GetTime() &&
                this->m_inputs == other.GetInputs() &&
                this->m_outputs == other.GetOutputs() &&
                this->m_extras == other.GetExtras());}
int CTransactionX::GetVersion() const {
        return this->m_version;}
int CTransactionX::GetLockTime() const {
        return this->m_lockTime;}
void CTransactionX::SetNull() {
        this->m_version = 1;
        this->m_inputs.clear();
        this->m_outputs.clear();
        this->m_lockTime = 0;}
bool CTransactionX::IsNull() const {
        return (this->m_inputs.empty() && this->m_outputs.empty());;}
string CTransactionX::ToString() const {
        return string();}
```

```
// Файл Main.cpp
#include "BlockChain.h"
#pragma warning(disable:4996)
int main(int argc, char* argv[]) {
	CBlockChain bChain = CBlockChain();
	cout << "Genesis..." << endl;
	bChain.AddGenesis();
	cout << "Mining block 1..." << endl;
	bChain.AddBlock(CBlock(1, "Block 1 Data"));
	cout << "Mining block 2..." << endl;
	bChain.AddBlock(CBlock(2, "Block 2 Data"));
	cout << "Mining block 3..." << endl;
	bChain.AddBlock(CBlock(3, "Block 3 Data"));
	system("pause");
	return 0; }
```

ДОДАТОК Б
ОПУБЛІКОВАНА ПРАЦЯ ЗА ТЕМОЮ ДИПЛОМУ

Тези на конференцію «Об'єднані наукою: перспективи міждисциплінарних досліджень» 2021

«ДОСЛІДЖЕННЯ ТЕХНОЛОГІЇ БЛОКЧЕЙН В КРИПТОВАЛЮТНІЙ ІНДУСТРІЇ»

У сучасному світі технологія блокчейн і ринок криптовалюти демонструють динамічний розвиток і передбачають розробку масштабних промислових додатків, здатних одночасно керувати багатьма процесами, обробляти і зберігати величезні масиви даних, забезпечуючи їх логічний взаємозв'язок і узгодженість. Можливості застосування технології блокчейн в бізнесі і промисловості не знають кордонів. Це підтверджується особливою увагою до цих питань з боку державних і приватних структур, високопрофесійних аналітиків та практиків й експертів.

Особливої популярності технологія набула, коли був перший різкий зріст ціни на криптовалюту «Біткоін». В той час багато людей почали входити в сферу та цікавитися новою перспективною технологією, в основі якої лежала ідея децентралізації, тому що «Біткоін» працює на основі цієї технології. Вагомим внеском у розвиток теоретичних основ блокчейн-технологій стала праця творця блокчейну Сатоші Накамото – «Біткоїн: цифрова пірингова система платежів», а також інші праці таких зарубіжних науковців, практиків та учасників системи блокчейн, як: Натаніель Поппер – книга «Цифрове золото», Майкл Кейсі – «Епоха криптовалют», Ден Шульман – «Блокчейн-революція» та інші. Ці вчені розглядали технологію з різних боків, а аналіз був доволі суттєвим ще з перших етапів існування блокчейну, що дало змогу зарубіжним компаніям швидко та ефективно використовувати її у своїй діяльності. Цю технологію досліджували такі вітчизняні вчені, як А. Усенко, Н. Ющенко, які в своїх роботах висвітлювали можливі перспективи розвитку технологій такого виду для української економіки.

Блокчейн – це технологія розподілених баз даних, яка базується на постійно зростаючому ланцюжку записів. Застосовувується в криптовалютах, а саме в технології біткоїна, оскільки дає змогу проводити ланцюгові транзакції з високим рівнем захисту від фальсифікації та підробки чи викрадення даних, а також високою швидкістю обробки операцій.

В загальному виді цю технологію можна представити як постійно зростаючу послідовність блоків, розподілених між учасниками. У кожен блок додається часова відмітка, які складаються в суворо визначений ланцюжок. Якщо спробувати змінити таку послідовність, то система відкине такий ланцюжок, оскільки послідовність буде визначено як неправильну. Для того щоб запобігти читанню правильної хеш-суми, технологія блокчейн використовує декілька засобів захисту, серед яких найважливішим є доказ роботи та доказ володіння. З цього слідує те, що учасники транзакцій не можуть обманути один одного, а дані є прозорими, оскільки наявна єдина база даних.

Отже, впровадження технології блокчейн лише набуває широкого розповсюдження серед компаній. Підприємства починають оцінювати вартість збереження власної інформації та забезпечення прозорості в побудові клієнтоорієнтованого підходу за допомогою високорівневих баз даних, таких як технологія блокчейн. Ті компанії, що почали впровадження ще на початкових етапах розвитку технології, мають можливість не лише використовувати наявний алгоритм роботи, але й модифікувати його, створюючи новий, більш розвинений продукт, який здатний конкурувати на ринку. Так, багато компаній стають постачальниками хмарних технологій блокчейн, що дає змогу отримувати досить високі прибутки та досліджувати нову нішу на ринку.