19. Karcheva, H.T., Ohorodnia, D.V. and Openko, V.A. (2017), "The digital economy and its impact on the development of national and international economics", *Finansovyi prostir*, no. 3 (27), p.13-21.
20. Norets, N.K. and Stankevich, A.A. (2017), "Digital Economy: State and Prospects of Development", *Innovatsionnyye klastery v tsifrovoy ekonomike: teoriya i praktika*, p.173-179.
21. Bell, D. (1974), The Coming of Post-Industrial Society: A Venure in Social Forecasting, Heinemann, Originally Published, London, UK.
22. Competing in the digital economy means owning the smart home – Accenture, available at: http://telecoms.com/480251/competing-in-the-digital-economy-meansowning-the-smart-home-accenture/ (access date May 15, 2018).
23. Shalaginov, A. (2017), "Smart City concept from "A" to "I", available at: http://infocom.uz/2017/02/18/koncepciya-smart-city-ot-a-do-ya/ (access date May 15, 2018).
24. Code4Health, available at: https://code4health.org/ *(access date May 15, 2018).*
25. Social networks in 2018: a global study, available at: https://www.web-canape.ru/business/socialnye-seti-v-2018-godu-globalnoe-issledovanie/ (access date May 15, 2018).
26. Top 15 of the most popular social networks and apps, available at: https://www.dreamgrow.com/top-15-most-popular-social-networking-sites/ (access date May 15, 2018).

Стаття надійшла до редакції 05.06.2018 р.

УДК 336.744
JEL Classification: F300

Бойко О.Г.,
аспірант[1] кафедри міжнародних фінансів
ДВНЗ «Київський національний економічний
університет імені Вадима Гетьмана»

АНАЛІЗ ТЕХНОЛОГІЧНИХ ІННОВАЦІЙ В СИСТЕМІ МІЖНАРОДНИХ РОЗРАХУНКІВ КРИПТОВАЛЮТОЮ

Boiko O.H.,
postgraduate student at the
department of international finance
Kyiv National Economic University
named after Vadym Hetman

ANALYSIS OF TECHNOLOGICAL INNOVATIONS IN A CRYPTOCURRENCY INTERNATIONAL PAYMENT SYSTEM

Постановка проблеми. Досліджуючи причини експансії криптографічної валюти в систему міжнародних розрахунків, зазвичай перелічуються технічні і економічні характеристики криптовалюти, залишаючи поза увагою технологічні характеристики Блокчейн – технологічної інновації XXI століття. Йдеться про міжнародну платіжну систему, яка надає анонімність користувачам, децентралізованість обліку та здійснення платіжних операцій, швидкі та цілодобово доступні платежі, визначеність грошової бази з емісією платіжних засобів на користь провайдерів платіжної системи. Однак дана платіжна система є стабільною та безпечною лише за дотримання низки доволі жорстких вимог з галузі інформаційної безпеки. Технічна сторона криптовалюти недостатньо розкрита для фахівців з економіки, які не мають глибоких знань в галузі телекомунікації інформації. Криптографічна валюта формально є відкритою для моніторингу та аналізу, але реально перевірити її безпечність та протестувати задекларовані принципи функціонування здається надскладним завдання, яке скоріш за все під силу лише IT-аудиторам. Останні зазначають, що наразі не існує стандартного способу перевірки бізнес-процесів, які базуються на Блокчейн-технології, оскільки такі процеси мають

[1] *Науковий керівник: Токар В.В. – доктор економічних наук, професор*

унікальну архітектуру та недостатньо стандартизовані. Як наслідок, кожне застосування технології відбувається лише після її прилаштування для конкретного використання. Існує недостаток розуміння та досвіду, пов'язаного з Блокчейн технологією, яка базується на сьогочасних, а не на історичних даних, котрі зазвичай є об'єктом аудиту [11].

Аналіз останніх досліджень і публікацій. Багато матеріалу, присвяченого криптографічним валютам, побудовано на примітивних прикладах із повсякденного життя, які хоча і сприяють засвоєнню матеріалу, але не дають справжнього розуміння криптовалюти. Деякі дослідники орієнтуються на читачів з високими навичками програмування, використовуючи приклади на певній мові програмування, наприклад Java Script. Вони демонструють конкретну імплементацію криптовалюти і пов'язують розрізненні елементи в одне ціле. Л. Хартіка використовує просту криптовалюту, яка має досить наївну, за словами автора, імплементацію, де лише назви методів відображають їх призначення, а їхній зміст не подається.

Американські дослідники Дж. Бонно, А. Міллер, Е. Фелтен, С. Голдфедер та А. Нараянан [9] детально аналізують технологію Блокчейн, на якій побудовані криптографічні системи міжнародних розрахунків. Дж. Кіерен описує Блокчейн в аналогії з системою управління версіями Git, в контексті бази даних SQL, децентралізованої бази даних Torrent і технології синхронізації даних Raft [7]. Використовуючи перелік вищезазначених технологій, наше дослідження описує їх разом з іншими техніками і технологіями, пов'язуючи їх в контексті моделювання криптографічної системи розрахунків. Значною мірою використано здобутки таких зарубіжних дослідників, як Ластер [8], Менезес [6] та Харрісон [3] для опису складових Блокчейн за аналогією. Результати проведеного аналізу останніх досліджень і публікацій підтверджують доцільність продовження вивчення поставленої проблеми, зокрема здійснити поглиблений аналіз технологічних інновацій в системі міжнародних розрахунків криптовалютою.

Постановка завдання. Метою статті є розкрити технологічні характеристики Блокчейн-технології для більш адекватної оцінки технічних та економічних характеристик платіжних систем, які на ній базуються. Для досягнення цієї мети пропонується використати метод аналогії: скласти перелік і описати релевантні для Блокчейн техніки та технології, а також змоделювати саму Блокчейн в категоріях обраних технологій.

***Виклад основного матеріалу дослідження*.** Описання технології Блокчейн за допомогою методу аналогії здійснюється в контексті: оборотньо-сальдової відомості, файлообмінного протоколу BitTorrent, системи управління версіями тестових файлів Git, техніки цифрового підпису, зв'язаного списку як цифрової структури збереження даних, хеш-функції, хеш-списку як цифрової структури збереження даних, запиту в базу даних SQL, консенсус алгоритму для вибору лідера RAFT. Посилання на ту чи іншу технологію здійснюється переважно для того, щоб сфокусувати увагу на певному завданні, яке цією технологією вирішується і абсолютно не обов'язково бути фахівцем в галузі, щоб зрозуміти основну її ідею. Спочатку окреслимо вищеназвані методики і технології, після чого спробуємо аналогічно проаналізувати технологію Блокчейн в їхньому контексті.

Хеш-список. У мові програмування Java зберігати дані можна не лише в простих структурах, як-от число, строка і т.д., а і в складніших. Пов'язані між собою складні структури для зберігання даних мають чотири типи: список (List), множина (Set), відповідність (Map) і черга (Queue), перші два з яких імплементують математичну концепцію "множини", а саме списком (List) називається сукупність елементів, в якому порядок має роль, в той час як в множині (Set) порядок елементів не має значення [2]. Блокчейн, що дослівно означає «ланцюг блоків», найкращим чином репрезентується як структура зберігання даних під назвою «Хеш-список» (Hash List), яка насправді не належить до доступних в Java структур. Хеш-список є зв'язаним списком, в створенні якого задіяна так звана хеш-функція.

Зв'язаний список. Розглянемо більш детально один із підтипів структури список, який називається зв'язаний список (Linked List) та зображений на Рис. 1.

Рис. 1. Зв'язаний список в програмуванні – структура даних, в якій елементи лінійно впорядковані не за номерами елементів, а за вказівниками, які входять в склад елементів списку та вказують на наступний та попередній елементи

Джерело: [4]

Для того, щоб репрезентувати зв'язаний список на якомусь прикладі, уявімо, що на комп'ютері створюється нова порожня папка \Папка1, в яку додаються чотири текстові файли Microsoft Word з назвами A.docx, Б.docx, B.docx і Г.docx, кожен з яких має кілька рядків з якоюсь інформацією. Завдання полягає в тому, щоб пов'язати файли A.docx, Б.docx, B.docx і Г.docx через список, тобто інша папка \Папка2 з цими ж самими чотирма файлами може інтерпретуватися як окремий список з елементами в іншій послідовності, наприклад Б.docx, B.docx, Г.docx і A.docx. Перший і останній елементи зв'язаного списку мають лише по одному посиланню. Для того, щоб лінійно впорядкувати файли від А до Г в \Папка1, необхідно додати до першого в списку файлу A.docx додаткову стрічку тексту, яка міститиме посилання (аналогічне до будь-якого посилання Інтернеті) на наступний елемент Б.docx. У файлі Б.docx має з'явитися дві стрічки тексту з посиланнями на попередній та наступний елементи, A.docx та B.docx відповідно. У файлі B.docx вказівник на попередній файл вказуватиме на Б.docx, а вказівник на наступний файл – на Г.docx. Файл Г.docx, будучи останнім елементом в зв'язаному списку \Папка1 матиме лише одне посилання на попередній елемент B.docx.

Таким чином, за допомогою посилань здійснюється створення зв'язаного списку, оскільки в \Папка1 гіпотетично могли б знаходитися інші файли, як-от Д.docx і E.docx, які будуть ігноруватися, оскільки існують поза посиланнями. Так само легко виключати вже існуючі елементи із зв'язаного списку, змінюючи посилання в попередньому та наступному елементах. Наприклад, файл B.docx буде виключено, якщо у файлі Б.docx стрічку з наступним вказівником змінити так, щоб він вказував на Г.docx, в той час як посилання на попередній елемент у файлі Г.docx поміняти з B.docx на Б.docx.

Хеш функція. Хеш-функція – це математична функція, яка використовує певний об'єкт (наприклад файл Microsoft Word у форматі .docs або повідомлення) як вхідний аргумент і повертає вихідним значенням хеш цього об'єкту. Наприклад, вихідне значення хеш-функції з вхідним аргументом A.docx виглядало б приблизно так: a11bef06a3f659402fe7563arf99ad00de2209e6. Хеш-функція повертає унікальний хеш для об'єкту і, в цілому, існує правило, що однаковий хеш можливий лише для однакових об'єктів.

Хеш-список. Отже, ідея концепції хеш-списку відрізняється від ідеї зв'язаного списку тим, що для кожного елемента рахується хеш, причому замість посилання на попередній елемент вказується хеш попереднього елементу. Таким чином, хеш попереднього елементу слугує додатковим вхідним аргументом для хеш-функції, яка крім самої інформації елементу використовує унікальну строку-назву попереднього елемента [9]. Таблиця 1 ілюструє дану структуру на прикладі чотирьох файлів A.docx, Б.docx, B.docx і Г.docx.

Таблиця 1

Хеш-список чотирьох файлів Microsoft Word, до яких включено хеш попереднього елементу в списку з метою впливу на розрахунок власного хешу файлу

A.docx	Б.docx	B.docx	Г.docx
1) Інформація. 2) Власний хеш.	1) Інформація. 2) Хеш попереднього файлу A.docx. 3) Власний хеш, порахований на основі (1) і (2).	1) Інформація; 2) Хеш попереднього файлу Б.docx. 3) Власний хеш, порахований на основі (1) і (2).	1) Інформація; 2) Хеш попереднього файлу B.docx; 3) Власний хеш, порахований на основі (1) і (2).

Джерело: розроблено автором

Внаслідок того, що два об'єкти є однаковими, коли вони мають однаковий хеш, видалення елемента з хеш-списку вимагає перерахунок хешів всіх наступних елементів. Це принципово відрізняється від зв'язаного списку, де видалення одного елемента списку лише вимагає корегування посилань в попередньому і наступного елементу. Натомість, в хеш-списку видалення/зміна інформації в документі Б.docx відразу стане помітною при валідації, оскільки власні хеші B.docx і Г.docx зміняться.

Git. Системи управління версіями тестових файлів Git широко застосовується не лише в програмуванні, а і в цілому при роботі з текстовими файлами, оскільки дозволяє зберігати скільки завгодно версій одного і того самого документу.

По-перше, системи управління версіями бувають традиційні (централізовані) і нетрадиційні (розподілені). У централізованій системі текстові файли зберігаються в репозитарії на сервері, і зміна файлу відбувається безпосередньо на сервері. Git належить до розподілених систем управління версіями, в яких централізований сервер хоча і існує, кожен користувач системи повністю локально копіює весь репозитарій. Таким чином, кожен користувач Git має як свою локальну копію репозитарію, так і спільний віддалений репозитарій, з яким час від часу синхронізується локальний репозитарій [8, с. 20]. Рис. 3 репрезентує взаємозв'язок в розподіленій системі.

Прикладом локального репозитарію може бути будь-яка папка на комп'ютері, наприклад \ЛокальнийРепозитарійПапки1, яка пов'язана з \Папка1, де знаходяться файли A.docx, Б.docx, B.docx і

Г.docx. Тобто зміна тексту в файлі A.docx відбувається на комп'ютері в робочому середовищі Microsoft Word, після чого змінений файл (A'.docx) додається до локального репозитарію. Щоб змінений файл A'.docx був доступний для інших користувачів Git, локальний репозитарій синхронізується з віддаленим репозитарієм. Іншими словами, змінений файл A'.docx спочатку збережеться в \Папка1, потім потрапить в \ЛокальнийРепозитарійПапки1, після чого буде відправлений на віддалений сервер.

По-друге, в Git застосовується зберігання стану файлів (snapshot storage), що нагадують фотознімок: після редагування тексту відбувається ніби фотографування файлів, причому можна зробити стільки знімків і тоді, коли це може бути найбільш доречно, наприклад, перед якоюсь великою зміною.

По-третє, для кожного такого знімку файлів розраховується хеш за допомогою хеш-функції та вхідних аргументів – файлів. Оскільки властивістю хеш-функції є те, що лише однакові об'єкти можуть мати однаковий хеш, то кожен знімок стану системи матиме унікальний хеш, тому що навіть несуттєва зміна в тексті робить файли неоднаковими.

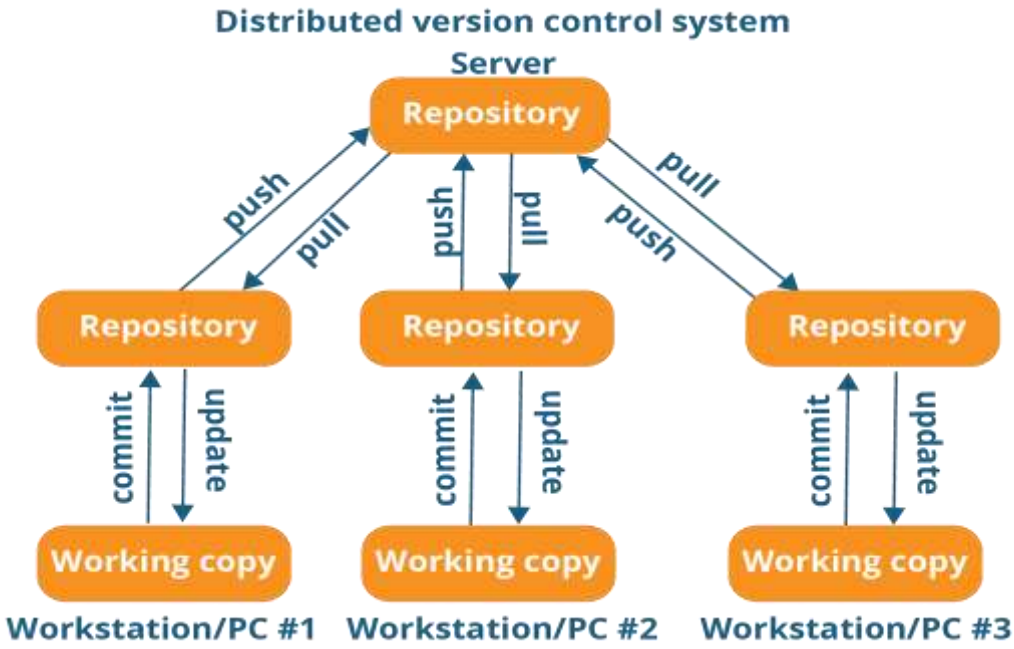

Рис. 3. Розподілена модель управління версіями текстових файлів, в якій зміни в файлах робляться не напряму, а через локальну копію центрального репозитарію. Зміни в файлах спочатку зберігаються локально, після чого дискретно (за рішенням користувача) синхронізуються з центральним сервером

Джерело: [8]

Таблиця 2

Ілюстрація кількох версій текстових файлів, які є результатом децентралізованої системи управління версіями Git. Дана таблиця представляє ту структуру, яка міститься на локальному та віддаленому репозитаріях, тобто не лише остання, а і всі попередні знімки системи є доступними

Хеш (скорочено)	Коментар	Файли
da500aa4f54	Змінена інформація в файлі A	A'.docx, Б.docx, B.docx і Г.docx.
cbf8f3eb47a1	Створено файли	A.docx, Б.docx, B.docx, Г.docx.

Джерело: розроблено автором

Зверніть увагу на те, що стан файлів з хеш da500aa4f54 має змінену версію файлу A'.docx. Найголовніше – це те, що попередній стан всієї системи зберігається з хеш cbf8f3eb47a1, який можна використати при необхідності.

Цифровий підпис. Криптологи визначають цифровий підпис як число, яке залежить від певного секрету та змісту підписуваного повідомлення. Секретом є закритий ключ, відомий лише тому, хто здійснює цифровий підпис. Популярною схемою цифрового підпису є ассиметричний цифровий підпис з додатком (asymmetric digital signatures scheme with an appendix). В цьому контексті "асиметричний" вказує на те, що вибирається закритий ключ і похідний від нього відкритий ключ. Закритий ключ, який зберігається в таємниці, використовується при підписанні повідомлень, тоді як відкритий ключ робиться загальнодоступним і використовується для валідації цифрових підписів даної особи.

"Додаток" в даному контексті означає, що криптологічна хеш-функція використовується для перетворення повідомлення в хеш і це є частиною алгоритму цифрового підпису [6, с. 37].

Концепція цифрового підпису вже застосовується при телекомунікації даних і також є невід'ємною частиною Блокчейн технології. Цифровий підпис надає базові криптологічні послуги, забезпечуючи:

– повноту даних (запевнення, що дані не були змінені нелегітимним або невідомим чином);

– автентифікацію походження даних (запевнення відповідності джерела даних задекларованому джерелу);

– неможливість заперечувати факти (запевнення, що особа не зможе заперечити факт попередньо здійснених дій та взятих зобов'язань) [6, с. 37].

Як зазначають дослідники, цифровий підпис схожий до звичайного підпису, оскільки забезпечує дві ключові вимоги. По-перше, лише ви можете зробити підпис, але будь-хто здатний його перевірити. По-друге, підпис прикріплено до конкретного документа, що робить неможливим використати колись зроблений підпис для іншого документа. Цифровий підпис базується на відкритому та закритому ключі та використовує три операції, а саме здійснюється генерація ключів, підпис повідомлення або в цілому інформації та валідація підпису. Дані операції схематично описуються трьома математичними функціями:

1. *(закритий ключ, відкритий ключ) = згенеруватиКлючі(довжина ключа).* Функція *згенеруватиКлючі* має один вхідний аргумент, що задає довжину двох вихідних значень функції – закритого та відкритого ключів, які використовуються для цифрового підпису.

2. підпис = підписати(закритий ключ, повідомлення). Функція *підписати* повертає цифровий підпис та має два вхідні аргументи – закритий ключ та власне повідомлення, яке підписується.

3. чи валідний підпис = валідувати(відкритий ключ, повідомлення, підпис). Функція *валідувати* повертає 1, якщо відкритий ключ, повідомлення та підпис дійсно валідні, або 0 у протилежному випадку [9, с. 15].

Таким чином, будь-хто з відкритим ключем може перевірити, чи певному повідомленню або інформації дійсно належить цифровий підпис. Тепер розглянемо імплементацію методів *згенеруватиКлючі, підписати* та *валідувати* в алгоритмі цифрового підпису (Digital Signature Algorithm) та конкретному прикладі.

Алгоритм цифрового підпису (DSA). У контексті описання цифрового підпису багато уваги приділяється відкритому і закритому ключам, тоді як поза увагою залишаються відкриті параметри, не знаючи яких взагалі неможливо перевірити достовірність цифрового підпису. До параметрів схеми цифрового підпису, які мають бути доступні разом з відкритим ключем, зазвичай, належать криптографічна хеш-функція та кілька чисел-параметрів. Для побудови системи цифрового підпису потрібно виконати наступні кроки:

1. Вибір криптографічної хеш-функції H(x).
2. Вибір простого числа q.
3. Вибір простого числа p, такого, що (p-1) ділиться на q.
4. Розрахунок числа g за формулою $g = 2^{(p-1)/q} \bmod p$.

Нагадаємо, що простим числом є натуральне (ціле позитивне число), що має рівно два різних натуральних дільники – одиницю і самого себе.

Генерація ключів представляє вибір числа $x \in (0, q)$ для закритого ключа та розрахунок іншого числа для відкритого ключа. Відкритий ключ *y* не вибирається незалежно, а насправді є функцією від закритого ключа *x,* яка в даному алгоритмі має вигляд $y = g^x \bmod p$. Оператор mod означає модулус числа, який дорівнює остатку при діленні чисел, наприклад 5 mod 3 = 2.

Відкритими параметрами є числа (*p, q, g, y*). Закритий параметр тільки один – число *x*. При цьому числа (*p, q, g*) можуть бути загальними для групи користувачів, а числа *x* і *y* є відповідно закритим і відкритим ключами конкретного користувача. При підписуванні повідомлення використовуються секретні числа *x* і *k*, причому число *к* має вибиратися випадковим чином при обчисленні підпису кожного наступного повідомлення. Оскільки *(p, q, g)* можуть бути використані для декількох користувачів, на практиці користувачів часто ділять за деякими критеріями на групи з однаковими параметрами (*p, q, g*), які тому називаються доменними параметрами [6].

Здійснення підпису здійснюється за наступним алгоритмом:

5. Вибір випадкового числа $k \in (0, q)$
6. Обчислення $r = (g^k \bmod p) \bmod q$. Вибір іншого k, якщо r = 0.
7. Обчислення $s = k^{-1}(H(\text{повідомлення}) + x \cdot r) \bmod q$. Вибір іншого k, якщо s = 0.
8. Підписом є пара (r, s).

Валідація цифрового підпису здійснюється за наступним алгоритмом:

1. Обчислення $w = s^{-1} \bmod q$.
2. Обчислення $u_1 = H(\text{повідомлення}) \cdot w \bmod q$.
3. Обчислення $u_2 = r \cdot w \bmod q$.

4. Обчислення $v = (g^{u_1} \cdot y^{u_2} \ mod \ p) \ mod \ q$.
5. Підпис дійсний, тільки якщо $v = r$.

Неважко бачити, що валідація цифрового підпису здійснюється через порівняння двох чисел *v* і *r*, причому *r i s* є числами з пари цифрового підпису. Також можна помітити, що для валідації необхідні відкритий ключ *y,* обрані параметри (*p, q, g*), хеш-функція H() та саме повідомлення, на якому ставиться цифровий підпис.

У 2013 році зловмисникам вдалося викрасти криптовалюту Біткоін з рахунків користувачів гаманців, які були написані для операційної системи Андроід. Як повідомляється, компонент гаманців, який відповідає за генерування випадкових чисел в алгоритмі цифрового підпису мав недолік, який призвів до того, що цифровий підпис стало можливо сфальсифікувати. Проблема полягала в алгоритмі цифрового підпису ECDSA (Elliptic Curve Digital Signature Algorithm), який, до речі, вважається безпечнішим за той, що розглядається в даному дослідженні. Не залежно від відкритого і закритого ключів, ECDSA вимагає, щоб випадкове число, яке використовується під час цифрового підпису використовувалося лише один раз. Якщо випадкове число використовується хоча б двічі, то закритий ключ, який асоціюється з відкритим ключем і генерується гаманцем, можна вирахувати [1].

Роль технології цифрового підпису в Блокчейн важко переоцінити, оскільки вміле використання елементів цифрового підпису робить платіжну систему дійсно децентралізованою. Для роботи системи, яка не використовує Блокчейн, потрібна база відповідності між реальними реквізитами автора (це може бути як приватна особа, так і організація) і відкритими ключами, а також всіма необхідними параметрами схеми цифрового підпису (хеш-функція, прості числа). Наприклад, подібною базою може служити центр сертифікації [6]. У криптологічній системі розрахунків реалізована ідея прирівняння відкритого ключа з компоненту цифрового підпису до реквізитів особи. Якщо подивитися на транзакцію, що має цифровий підпис, то відкритий ключ цієї транзакції буде розглядатися як особа, яка здійснює транзакцію. Для користувача стати особою в Блокчейн означає володіти закритим ключем, що асоціюється з відкритим ключем. Як наслідок, для того, щоб користувач отримав нові реквізити, достатньо згенерувати пару відкритого і закритого ключів в системі цифрового підпису [9, с. 19].

База даних SQL. Блокчейн є базою даних, яку також можна порівняти з базою даних SQL, оскільки обидві використовують таблицю для зберігання даних. Для даного дослідження релевантними командами в SQL є транзакція (transaction), вираз (expression) та тріггер (trigger), за допомогою яких дані додаються до таблиці в базі даних SQL [7].

Вираз в SQL – це комбінація констант, змінних та операторів, що призводить до певного значення. Тріггер в SQL – це програма, що зберігається в базі даних і запускається всякий раз, коли до таблиці додаються, обновлюються або видаляються дані. Наприклад, таблиця, в якій ведеться реєстр продажів, може мати тріггер, який перевіряє вартість продажу і застосовує знижку, якщо вартість угоди перевищує певне порогове значення. Транзакція в SQL – це набір з однієї або більше команд, які логічно поєднані і мають або повністю бути застосовані до таблиці з даними, або бути повністю відхиленими. Тобто застосування лише частини команд не допускається, оскільки діє принцип "все або нічого". Наприклад, грошовий трансфер з одного рахунку на інший має дві команди: зменшення коштів на одному рахунку та збільшення коштів на іншому [3].

У цілому, база даних має відповідати таким чотирьом вимогам (так званий ACID principle):

1. Неподільність транзакцій (Atomic) – до бази даних застосовуються або всі, або жодні команди транзакції.
2. Закінченість бази даних (Consistent) – база даних є закінченою як до, так і після транзакцій.
3. Ізольованість (Isolated) – при виконанні багатьох транзакцій водночас, одночасність виконання однієї транзакції не повинна мати впливу на інші.
4. Надійність (Durable) – коли транзакція збережена в таблиці, то пов’язана з транзакцією зміна має зберегтися [3, с. 179].

Принциповою рисою бази даних SQL є те, що кожна транзакція невідкладно призводить до додавання, зміни або видалення файлів з бази даних. Іншими словами, групування транзакцій не відбувається, оскільки кожна команда, яка починає транзакцію неминуче закінчується командою, яка зберігає зміни [3, с. 182].

Уявімо, що згадуваний раніше файл A.docx містить таблицю з реєстром грошових переказів і додавання нового грошового переказу здійснюється за допомогою SQL транзакції. Така транзакція має містити не лише рахунок відправника і отримувача, а і кілька валідаційних операцій.

Таким чином, будучи невід’ємною частиною бази даних, валідація транзакцій в SQL здійснюється при кожній транзакції.

Файлообмінний протокол BitTorrent. В децентралізованій системі обмін інформацією здійснюється децентралізовано, тобто безпосередньо між учасниками без центрального сервера. Метою даного протоколу є полегшення обміну файлами великих розмірів у мережі [5]. BitTorrent належить до децентралізованих систем і проілюстрований на рис. 4.

Реєстр транзакцій. Транзакції бувають двох типів: платіжні та емісійні. Платіжні транзакції містять адреси отримувачів, адреси відправників, номінал, цифрові підписи передавачів. Емісійні

транзакції не мають адрес та цифрових підписів відправників. Блок складається з даних про транзакції, вказівника на хеш попереднього блоку та їх хешу.

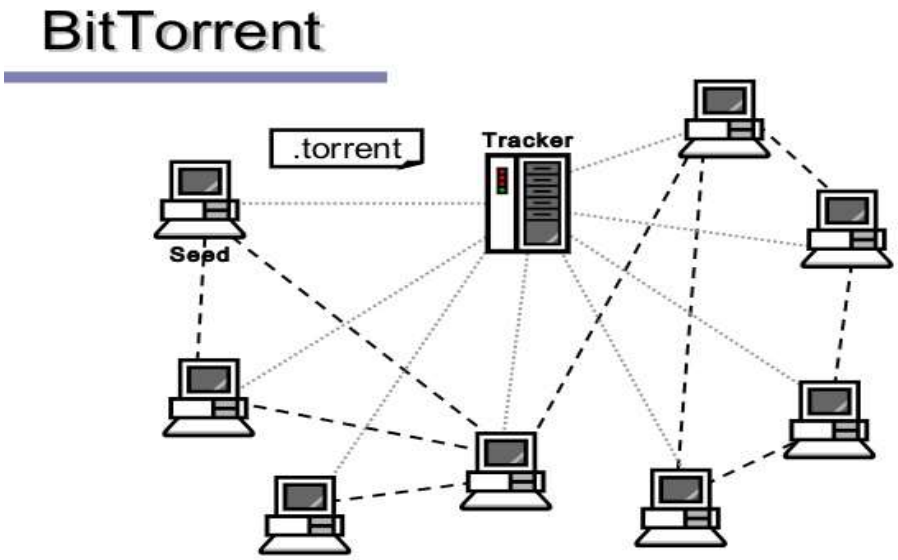

Рис. 4. Пірингова система BitTorrent, в якій обмін файлами здійснюється безпосередньо між учасниками. Центральний сервер (трекер) використовується тільки для того, щоб клієнти могли знайти один одного

Джерело: [5]

При платіжній транзакції відбувається знищення одних монет і створення інших. Обов'язковими умовами валідності платіжної транзакції є наступні: монети, які знищуються, мали бути створенні в попередніх транзакціях; монети, які знищуються, не мали бути спожиті в попередніх транзакціях; номінал монет, які знищуються, має дорівнювати номіналу створених монет; транзакції підтверджуються власниками монет, які знищуються, за допомогою цифрового підпису [9, с. 24].

Блокчейн-транзакції можна інтерпретувати в контексті оборотно-сальдової відомості, в якій відбувається кредитування одних бухгалтерських рахунків та дебетування інших. За такого підходу критерії валідності транзакцій набудуть наступний вигляд:

– субрахунки, які кредитуються, мають позитивне значення по дебету;

– субрахунки, які кредитуються, не кредитувалися в попередніх транзакціях;

– номінал, на який кредитується один рахунок має дорівнювати номіналу, на який дебетується інший рахунок (інші рахунки);

– для валідності транзакцій необхідна згода власників всіх рахунків, які кредитуються.

Нагадаємо, що рахунки бувають дебетні (інколи називаються активними рахунками) і кредитні (інколи називаються пасивними рахунками). Дебетний рахунок збільшується за дебетом, як, наприклад, «Рахунки в банках» згідно класу 31 в плані рахунків ПСБО, а кредитний рахунок збільшується за кредитом, як-от «Інший операційний дохід» згідно класу 71 в плані рахунків ПСБО.

Емісійна транзакція (coinbase transaction) – це транзакція, за допомогою якої відбувається емісія криптовалюти. Емісійна транзакція відрізняється від платіжної тим, що створення монет не супроводжується знищенням інших монет [9, с. 65].

Кожен блок в Блокчейн відображає стан системи, який можна порівняти зі спрощеною оборотно-сальдовою відомістю. Перше обмеження такої уявної оборотно-сальдової відомості полягає в тому, що в ній існує всього один дебетний і один кредитний рахунок.

Також уявімо, що субрахунками дебетного (активного) рахунку є Блокчейн-адреси (хеш відкритих ключів), враховуючи вищенаведене обмеженням, згідно якого кожен субрахунок може дебетуватися і кредитуватися всього один раз на одну і ту ж саму суму. Як наслідок, субрахунки, в яких вже відбулося нарахування по дебету та списання по кредиту автоматично матимуть нульове сальдо, і не можуть більше використовуватися.

Через те, що існують всього два види транзакцій, емісійна і платіжна, за аналогією можна визначити, що в оборотно-сальдовій відомості можливі лише два відповідні бухгалтерські проведення, наприклад:

- Емісійна транзакція: Дт «Рахунки в банках» субрахунок 1 – Кт «Інший операційний дохід».

• Платіжна транзакція: Дт «Рахунки в банках» субрахунок 2 – Кт «Рахунки в банках» субрахунок 1.

Таким чином, якщо розглядати криптографічну валюту з точки зору активу, вона є дебетовим сальдо на певному субрахунку, семантика якого визначається Блокчейн-адресою – хеш-функцією відкритого ключа, який є невід'ємною частиною цифрового підпису. Через те, що відкритий ключ використовується як субрахунок, ініціювати платіжну транзакцію можливо за умови володіння відповідним йому закритим ключем, яким підписується транзакція. У даному контексті ключовим є облік вже здійснених транзакцій, який здійснюється в Блокчейн децентралізовано.

Консенсус алгоритм для вибору лідера RAFT. Необхідність розгляду даної технології продиктована тим, що Блокчейн є децентралізованою базою даних, що локально знаходиться на нодах, які доповнюють базу даних новими записами. Виходячи з цього, важливе місце займає консенсус-алгоритм, який надає дозвіл тому чи іншому ноду додавати дані в базу даних так, щоб це було прийнятно для інших нодів.

Консенсус алгоритми дозволяють багатьом машинам працювати як цілісна група, яка може вистояти невдачі деяких її членів. Технологія Raft охоплює питання вибору лідера, який стає відповідальний за додавання даних до бази даних та регламентує зміну лідера у випадку невдачі. Головне завдання Raft – забезпечити безпеку та функціональність бази даних при проблемах, пов'язаних з передачею інформації, таких як затримки, втрата, дуплікація і пересортування інформації. Raft реалізує консенсус таким чином, що спочатку вибирається визначений лідер, якому надається повна відповідальність в управлінні базою даних. Маючи лідера, спрощує управління базою даних, оскільки лідер може вирішити, як і де розміщувати нові записи, не консультуючись з іншими нодами і інформаціє перетікає від лідера до інших машин. Однак лідер може зламатися або бути відключеним від інших машин, в такому випадку Raft назначає іншого лідера. Важливим є те, що новий лідер вибирається випадковим чином, тобто рандомно. Крім того, лідер може лише додавати нову інформацію до бази даних, а змінювати вже наявну – ні [10, с. 1–4].

Вибір лідера в Блокчейн вирішується інакше, ніж в Raft. В Raft лідер (тобто нод, який отримав право додавати інформацію в базу даних) визначається випадковим чином, тоді як в Блокчейн визначення лідера часто відбувається пропорційно до обчислювальної потужності ноду (так званий консенсус-алгоритм proof-of-work) [7]. Наступний приклад демонструє консенсус-алгоритм Raft. Нехай три користувача О, В і К складають систему, що забезпечує функціональність бази даних, яка складається з Microsoft Word файлів. Припустимо, що лідером є К, який створює файл А.docs і транслює його іншим користувачам О і В. Далі К виходить із системи, і Raft випадковим чином вибирає з-поміж О і В нового лідера, який і буде здійснювати додавання файлів до бази даних, створюючи і передаючи В наступні файли Б.docs, В.docs і Г.docs.

Моделювання криптографічної системи розрахунків аналогічно до вищенаведених технік і технологій. Нехай О, В, К і Н користуються криптографічною системою розрахунків на основі Блокчейн, в якій визначені доменні параметри та тип хеш-функції. Ці чотири особи здійснюють час від часу платіжні транзакції, надсилаючи один одному монети. Припустимо також, що О, В і К додатково здійснюються майнинг, тому на їх користь також відбуваються емісійні транзакції. У цьому прикладі моделюється додавання до вже існуючого Блокчейн двох блоків, які відображають наступні події: О має одну монету, яку він надсилає Н; К є лідером, він валідує транзакцію, формує блок і додає його до Блокчейн; емісійна транзакція номіналом в 25 монет здійснюється на його користь; О і В отримують новий блок і включають його до своїх локально збережених Блокчейн.

Вищезазначені події знаходять наступне описання в запропонованій системі понять. Припустимо, що О зберігає 1 монету на рахунку під назвою (тобто хешем відкритого ключа) ХВК0001. Оскільки О має закритий ключ, який було використано при генерації відкритого ключа з хешем ХВК0001, то О може підписати платіжну транзакцію номіналом 1 монета на користь Н, яка буде валідною. Н генерує в цій криптографічній системі нову пару ключів, закритий і відкритий ключ, і нехай хеш відкритого ключа становить ХВК0002. Очевидно, що в ХВК0002 є рахунком в Блокчейн, з яким поки що не асоційовано жодних монет. Далі О записує цю платіжну транзакцію в бухгалтерській проводці. О використовує закритий ключ для ХВК0002 і бухгалтерську проводку для цифрового підпису, який наприклад становив пару цифр (4,9).

Після цього платіжна транзакція транслюється О в систему і О, В та К отримують її і здійснюють валідацію. Валідація включає, по-перше, перевірку того, що 1 монета була створена і не була спожита в попередніх транзакціях і що номінал на ХВК0002 має дорівнювати номіналу на ХВК0001. По-друге, це також перевірка цифрового підпису з використанням відкритого ключа, що хешується до ХВК0002, проводки, і цифрового підпису (4,9). Кожен з майнерів має локальну копію всіх попередніх проводок, що також згруповані в блоки, які мають форму цифрової структури даних хеш-ліст. Ця структура даних забезпечує неможливість зміни історичних транзакцій без того, щоб це не стало помітно і не призвело до зміни хешу всіх наступних блоків після зміненого блоку. Оскільки більше транзакцій не відбулося, то К (який наразі є лідером) формує блок, за що на його користь включається емісійна транзакція. Щоб отримати емісійні монети, К генерує відкритий і закритий ключі, і нехай відкритий ключ хешується до ХВК0003 (таблиця 3).

Таблиця 3

Зміст блоку, який додається до Блокчейн

Хеш блок ХБ000N					
№	Тип транзакції	Дт рахунку	Кт рахунку	Номінал	Підпис
1.	Платіжна	ХВК0002	ХВК0001	1 монета	(4,9)
2.	Емісійна	ХВК0003	ХХХ Дохід від майнингу	25 монет	-

Джерело: розроблено автором

Потім К локально додає новостворений блок до Блокчейн. Іншими словами, К робить Git-коміт блоку в локальний репозитарій. Контент блоку і хеш попереднього блоку будуть використані для генерації хешу вже всього блоку, який нехай буде ХБ000N. Нарешті, ХБ000N гарантує незмінність двох проводок, оскільки будь-яка зміна призведе до зміни ХБ000N, що не може залишитися непомітним для майнерів О, В і навіть звичайного користувача Н. Після чого передає свою локальну версію репозитарію майнерам О і В, які додають новий блок до своїх локальних Блокчейн репозитаріїв.

Висновки з проведеного дослідження. Отже, дане дослідження демонструє, що криптовалюту слід розглядати з технічної точки зору. На нашу думку, проведені аналогії дозволяють зрозуміло описувати потік транзакцій, який відбувається в типовій децентралізованій системі міжнародних розрахунків. Результати дослідження демонструють можливість до значної варіативності в системах Блокчейн, оскільки вони мають багато вузлів, кожен з яких представлений окремою технологією. Прикладом є алгоритми вибору лідера, правил валідації транзакцій, цифрового підпису з його хеш-функцією і доменними параметрами. Хоча можливість варіювати ту чи іншу технологію призводить до складності розуміння конкретної криптографічної системи розрахунків, даний аналіз може полегшити розуміння всієї системи в цілому.

Література

1. Chirgwin R. Android bug batters Bitcoin wallets. Old flaw, new problem. URL: https://www.theregister.co.uk/2013/08/12/android_bug_batters_bitcoin_wallets/ (дата звернення: 06.05.2018).

2. CodeJava. Overview of Java Collections Framework API (UML diagram). URL: http://www.codejava.net/java-core/collections/overview-of-java-collections-framework-api-uml-diagram (дата звернення: 14.05.2018).

3. Harrison G., Feuerstein S. MySQL stored procedure programming. 1 Iss. Sebastopol CA: O'Reilly, 2006. xxiii, 609.

4. Jay B. Doubly Linked List Implementation Guide. URL: https://www.thecodingdelight.com/doubly-linked-list/ (дата звернення: 19.05.2018).

5. Johnsen, J. A., Karlsen, L.E., Birkeland, S.S. Peer-to-peer networking with BitTorrent. URL: http://web.cs.ucla.edu/classes/cs217/05BitTorrent.pdf (дата звернення: 23.05.2018).

6. Johnson D., Menezes A., Vanstone S. The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security*. 2001. Vol. 1. iss. 1. p. 36–63.

7. Kieren J. L. Blockchains by analogies and applications: How blockchain compares to Git, Raft, and other technologies. URL: https://www.oreilly.com/ideas/blockchains-by-analogies-and-applications (дата звернення: 05.05.2018).

8. Laster B. Professional git. Indianapolis IN: John Wiley and Sons, 2016. 454 p.

9. Narayanan A. Bitcoin and cryptocurrency technologies: A comprehensive introduction. Princeton: Princeton University Press, 2016. 304 p.

10. Ongaro D., Ousterhout J. In Search of an Understandable Consensus Algorithm (Extended Version). URL: https://raft.github.io/raft.pdf (дата звернення: 14.05.2018).

11. PwC New Ventures. PwC Blockchain Validation Solution. URL: https://www.pwc.com/us/en/about-us/new-ventures/pwc-blockchain-validation-solution.html (дата звернення: 03.05.2018).

References

1. Chirgwin, R. Android bug batters Bitcoin wallets. Old flaw, new problem, available at: https://www.theregister.co.uk/2013/08/12/android_bug_batters_bitcoin_wallets/ (access date: 06.05.2018).

2. CodeJava. Overview of Java Collections Framework API (UML diagram), available at: http://www.codejava.net/java-core/collections/overview-of-java-collections-framework-api-uml-diagram (access date: 14.05.2018).

3. Harrison, G. and Feuerstein, S. (2006), MySQL stored procedure programming. 1st Iss. Sebastopol CA: O'Reilly, xxiii, 609.

4. Jay, B. Doubly Linked List Implementation Guide, available at: https://www.thecodingdelight.com/doubly-linked-list/ (access date: 19.05.2018).

5. Johnsen, J.A., Karlsen, L.E. and Birkeland, S.S. Peer-to-peer networking with BitTorrent, available at: http://web.cs.ucla.edu/classes/cs217/05BitTorrent.pdf (access date: 23.05.2018).

6. Johnson D., Menezes A., Vanstone S. (2001), The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security*, Vol. 1, iss. 1, pp. 36–63.

7. Kieren, J.L. Blockchains by analogies and applications: How blockchain compares to Git, Raft, and other technologies, available at: https://www.oreilly.com/ideas/blockchains-by-analogies-and-applications (access date: 05.05.2018).

8. Laster, B. (2016), Professional git. Indianapolis IN: John Wiley and Sons, 454 p.

9. Narayanan, A. (2016), Bitcoin and cryptocurrency technologies: A comprehensive introduction. Princeton: Princeton University Press, 304 p.

10. Ongaro, D. and Ousterhout, J. In Search of an Understandable Consensus Algorithm (Extended Version), available at: https://raft.github.io/raft.pdf (access date: 14.05.2018).

11. PwC New Ventures. PwC Blockchain Validation Solution, available at: https://www.pwc.com/us/en/about-us/new-ventures/pwc-blockchain-validation-solution.html (access date: 03.05.2018).

12. Boiko, O. (2018), "Expansion of the crypto-currency into the system of international payments under the influence of Blockchain technology: evidence and reasons", *Hlobalni ta natsionalni problemy ekonomiky,* no. 22, pp. 31–38.

Стаття надійшла до редакції 12.06.2018 р.

Рецензент: д.е.н., професор ДВНЗ «Київський національний економічний університет імені Вадима Гетьмана» В.В.Токар