МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри Комп'ютеризованих систем захисту інформації

_____ Михайло СТЕПАНОВ
«____»_____ 2023 р.

На правах рукопису
УДК 004.056.5:004.4

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Програмний модуль захисту даних в інформаційних системах технологією блокчейн

Виконавець: Олексій САВЮК
Керівник: к.т.н. Наталія ГУЛАК
Консультант розділу «Охорона навколишнього середовища»: к.т.н., доцент Тетяна ДМИТРУХА
Нормоконтролер: к.т.н. Наталія ГУЛАК

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет: Кібербезпеки та програмної інженерії

Кафедра: Комп'ютеризованих систем захисту інформації

Освітній ступінь: Магістр

Спеціальність: 125 «Кібербезпека»

Освітньо-професійна програма: «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
Завідувач кафедри Комп'ютеризованих систем захисту інформації

_____ Михайло СТЕПАНОВ
«__» _____ 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи
здобувача вищої освіти Савюка Олексія Васильовича

1. Тема: *Програмний модуль захисту даних в інформаційних системах технологією блокчейн*
затверджена наказом ректора від «15» вересня 2023 р. № 1814/ст.
2. Термін виконання: з 16.10.2023 р. по 31.12.2023 р.
3. Вихідні дані: нормативно-правові документи України з кібербезпеки, криптографічні методи, методи шифрування, криптографічні алгоритми, хеш-функції, технологія блокчейн, мова програмування C#.
4. Зміст пояснювальної записки: проведення порівняльного аналізу за класифікацією методів захисту даних; аналіз методів захисту баз даних технологією блокчейн; розробка та тестування програмного модуля захисту даних в ІС на основі технології блокчейн.

5. КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

№ з/п	Етапи виконання кваліфікаційної роботи	Термін виконання етапів	Примітка
1.	Уточнення постановки задачі	16.10.2023	*Виконано*
2.	Аналіз літературних джерел	23.10.2023	*Виконано*
3.	Обґрунтування вибору рішення	30.10.2023	*Виконано*
4.	Збір інформації	07.11.2023	*Виконано*
5.	Проведення аналізу за класифікацією	14.11.2023	*Виконано*
6.	Аналіз методів захисту баз даних	19.11.2023	*Виконано*
7.	Розробка та тестування програмного модуля	26.11.2023	*Виконано*
8.	Глобальна кліматична проблема	03.12.2023	*Виконано*
9.	Апробація роботи на «The V International Science Conference «Trends in science regarding the creation of new teaching methods»	08.12.2023	*Виконано*
10.	Перевірка на антиплагіат	11.12.2023	*Виконано*
11.	Оформлення і друк пояснювальної записки	15.12.2023	*Виконано*
12.	Оформлення презентації	19.12.2023	*Виконано*
13.	Отримання рецензій від рецензента	22.12.2023	*Виконано*

6. Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона навколишнього середовища	Дмитруха Т.І.		

7. Дата видачі завдання: «16» жовтня 2023 р.

Здобувач вищої освіти Олексій САВЮК

(підпис, дата)

Керівник кваліфікаційної роботи Наталія ГУЛАК

(підпис, дата)

РЕФЕРАТ

Кваліфікаційна робота на тему: «Програмний модуль захисту даних в інформаційних системах технологією блокчейн» складається зі вступу, основної частини, що містить 4 розділи, 4 висновки до кожного розділу, загального висновку, 2 додатків та списку використаної літератури. Загальний обсяг роботи – 94 сторінки. Робота містить 24 рисунка та 3 таблиці. Список використаних джерел включає 32 джерела.

Метою роботи є розробка та тестування програмного модуля захисту інформації в базах даних на основі технології блокчейн мовою програмування C#.

В роботі розроблено та протестовано програмний модуль для захисту інформації в базах даних на основі технології блокчейн мовою програмування C#.

Розроблений метод та програмне забезпечення відносяться до галузі інформаційної безпеки і можуть бути використані для підвищення рівня захищеності.

Можливі напрямки розвитку цієї роботи пов’язані із розширенням моделі технології блокчейн.

Ключові слова: інформаційна безпека, захист інформації, захист даних, блокчейн, технологія блокчейн, серверний блокчейн, бази даних, захист баз даних.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ІТ	інформаційні технології;
ІС	інформаційні системи;
БД	база даних;
ВД	вихідні дані;
ІБ	інформаційна безпека;
ІКС	інформаційно-комунікаційна система;
ІППР	інтегроване представлення параметрів ризику;
ІР	інформаційний ресурс;
ЗІ	захист інформації;
КС	комп’ютерна адреса;
ЛЗ	лінгвістична змінна;
НЗ	нечітка змінна;
НЛ	нечітка логіка;
НМ	нечітка множина;
ПЗ	програмний засіб;
ПК	персональний комп’ютер;

ВСТУП

Актуальність. За останні роки, багато розбробників ІТ сфери стали все більше приділяти уваги новій технлогії. Вслід за інтернетом надвигається нова інформаційно-технічна хвиля, одним з найважливіших компонентів якої є технологія блокчейну. Цю технологію, яка складається з ланцюга блоків, все частіше починають називати революцією в збереженні та розподіленні обработаної інформації[1].

Технологія блокчейн і ринок криптовалюти демонструють динамічний розвиток і привертають пильну увагу. Нововведення цієї технології полягає в тому, що інформація про транзакції більш не зберігається в централізованій базі даних, а передається на комп'ютери всіх учасників мережі, які зберігають дані локально[1].

Bitcoin, перший у світі блокчейн, що є втіленням цієї концепції, був запущений 3 січня 2009 року та успішно функціонує вже майже 10 років.

Інформацію про транзакції зберігають у вигляді "ланцюжка блоків", в кожному з яких фіксується певна кількість комунікацій. Кожна транзакція в блокчейні – це дані, які подальше перевіряються незалежними учасниками і вписуються в глобальну історію транзакцій. По суті, технологія блокчейн є універсальним засобом зберігання та обробки інформації у практично будь-якій галузі діяльності[1].

Революційний характер блокчейну становить загрозу компаніям, що надають технологічні послуги в різних секторах економіки, таких як фінанси, енергетика, охорона здоров'я та сільське господарство. Впровадження блокчейну сприяє переходу до нових бізнес-моделей та оптимізації бізнес-процесів[1].

Наприклад, компанії, які надають фінансові послуги, застосовуючи цю технологію для підвищення достовірності фінансових транзакцій і зниження

витрат, можуть стимулювати інших постачальників впроваджувати платформи та рішення на основі блокчейн.

Проте, якщо постачальники пропустять цю можливість і не впровадять технологію вчасно, то цим можуть скористатися безліч блокчейн-стартапів.

Так само як економіка Інтернету, блокчейн створить нову економіку, і ми не повинні забувати про цей потенціал. Криптотехнологічна економіка буде економікою, заснованою на децентралізованій довірі як у політичному плані, так і в плані цифрової архітектури. Блокчейн забезпечить усім доступ і зменшить висоту бар'єрів всім учасників[1].

Поширення та обмін інформацією – ніша, спочатку зайнята Інтернетом, тоді як функцією блокчейна є передача цінностей. Ось основна суть того, що вам слід знати про блокчейн, і майже все, що випливає з базової ідеї. Незважаючи на породжене ним сумління і занепокоєння, ми повинні пам'ятати, що, по суті, блокчейн – це перспективна технологія.

Технологія блокчейн швидко розвивається і модернізується, тому тема дипломної роботи є актуальною.

Метою дипломної роботи є розробка та тестування програмного модуля захисту інформації в базах даних на основі технології блокчейн мовою програмування C#.

Для досягнення поставленої мети у роботі вирішуються такі **задачі:**

- провести порівняльний аналіз за класифікацією методів захисту даних;
- аналіз методів захисту баз даних технологією блокчейн;
- розробка та тестування програмного модуля захисту даних в ІС на основі технології блокчейн.

Галузь застосування: Розроблений метод та програмне забезпечення відносяться до галузі інформаційної безпеки і можуть бути використані для

підвищення рівня захищеності баз даних в ІС за рахунок аутентифікації та авторизації.

Об'єктом дослідження є процес захисту інформації баз даних в ІС.

Предметом дослідження є методи захисту інформації баз даних від несанкціонованого доступу та технологія блокчейн.

Методи дослідження базуються на аналізі методів захисту інформації в базах даних технології блокчейн та об’єктно-орієнтованого програмування (для програмної реалізації розробленого методу).

Новизна одержаних результатів полягає в наступному:

- на основі розробленої централізованої серверної бази даних було розроблено захист від змінення інформації у базі даних за рахунок розподілення прав доступу користувачів при використанні технології блокчейн, що дало можливість надавати користувачам інформацію обмеженої частки бази даних.

Практична цінність отриманих результатів:

- розроблений програмний продукт мовою програмування C# удосконалює програмний модуль для захисту даних технологією блокчейн за рахунок введення аутентифікації та авторизації.

Апробація. Основні положення роботи доповідалися та обговорювалися на таких конференціях:

- Савюк О. В. Захист даних технологією блокчейн /Н.К. Гулак, О.В. Савюк //V International Scientific and Practical Conference "Trends in science regarding the creation of new teaching methods", October 16-18, 2023, Madrid, Spain. - С. 197 - 199.

РОЗДІЛ 1. АНАЛІЗ ТЕХНОЛОГІЙ І МЕТОДІВ ЗАХИСТУ ДАНИХ НА ОСНОВІ НОРМАТИВНОЇ БАЗИ УКРАЇНИ З КІБЕРБЕЗПЕКИ

1.1. Правове забезпечення кібербезпеки в інформаційних системах

Розглянемо правові аспекти забезпечення кіберпезпеки в інформаційних системах:

1.1. Закон України "Про захист інформації в інформаційно-телекомунікаційних системах"[2] був прийнятий у 1992 році і декілька разів змінювався та доповнювався з метою забезпечення безпеки та захисту інформації в інформаційно-телекомунікаційних системах[2].

Він регулює роботу інформаційних систем, які використовуються в урядових структурах, комерційних підприємствах, та інших сферах.

Цей закон передбачає створення спеціальних органів та інфраструктури для забезпечення інформаційної безпеки. Важливою є роль Національного центру кібербезпеки України, який відповідає за координацію та моніторинг кіберзагроз та інцидентів[2].

Закон також передбачає застосування сучасних технологій, таких як криптографія, електронний підпис, ідентифікація користувачів, для забезпечення безпеки інформації. Важливою є роль суб'єктів інформаційних відносин у додержанні стандартів і правил, які передбачені цим законом[2].

Цей закон має важливе значення для забезпечення безпеки інформації в Україні та регулювання інформаційних відносин у сучасному цифровому світі:

1. Класифікація інформації: Закон регулює класифікацію інформації на відкриту, обмежену для внутрішнього використання та конфіденційну[2];

2. Заходи захисту: Закон передбачає встановлення технічних та організаційних заходів для забезпечення інформаційної безпеки, включаючи шифрування, автентифікацію, контроль доступу тощо[2];

3. Аудит і контроль: Закон вимагає проведення аудитів і контролю за дотриманням правил забезпечення інформаційної безпеки у підприємствах та органах влади[2];

4. Відповідальність: Він визначає відповідальність за порушення правил інформаційної безпеки, включаючи штрафи та кримінальну відповідальність[2];

5. Міжнародне співробітництво: Закон визначає можливість співробітництва з іншими країнами для обміну інформацією та спільного захисту інформації[2].

1.2. Державний стандарт України ДСТУ 3396.0-96 "Захист інформації. Технічний захист інформації. Загальні положення"[3] - це один із стандартів, що регулює аспекти захисту інформації в інформаційних системах в Україні.

"Загальні положення" встановлює загальні вимоги та принципи захисту інформації в інформаційних системах. Його ціль - забезпечення конфіденційності, цілісності та доступності інформації[3].

Стандарт містить важливі вказівки для організацій та підприємств, які використовують інформаційні системи, стосовно технічних засобів захисту інформації. Він надає рекомендації щодо класифікації інформації, контролю доступу, використання криптографії, аутентифікації та контролю конфігурації системи[3].

Цей стандарт є важливим інструментом для забезпечення інформаційної безпеки в Україні, особливо в контексті зростаючих загроз кібербезпеці та захисту даних. Він допомагає організаціям вдосконалити свої інформаційні системи та дотримуватися вимог до технічного захисту інформації[3].

Основні положення стандарту включають:

1. *Загальні вимоги до захисту інформації:* Стандарт встановлює загальні вимоги до технічного захисту інформації, включаючи заходи для забезпечення конфіденційності, цілісності та доступності інформації[3];

2. *Класифікація інформації:* Стандарт розглядає класифікацію інформації на відкриту, обмежену та конфіденційну та надає вказівки стосовно рівнів захисту для кожного класу інформації[3];

3. *Технічні засоби захисту:* ДСТУ 3396.0-96 визначає вимоги до використання технічних засобів захисту інформації, таких як шифрування, ідентифікація, контроль доступу та інші[3];

4. *Контроль доступу:* Стандарт розглядає важливі аспекти контролю доступу до інформації, включаючи встановлення прав доступу та аутентифікацію користувачів[3];

5. *Конфігурація системи і захист даних:* Стандарт регулює конфігурацію інформаційних систем, зокрема, заходи щодо захисту даних під час їх обробки, зберігання та передачі[3];

6. *Моніторинг та аудит:* Він вимагає проведення моніторингу та аудиту системи захисту інформації з метою виявлення порушень та інцидентів[3].

1.3. ISO/IEC 27001[4] - це міжнародний стандарт, який визначає вимоги для створення та управління системами управління інформаційною безпекою в організаціях. Його основна мета - забезпечити конфіденційність, цілісність та доступність інформації та захистити її від ризиків та загроз[4].

Стандарт допомагає організаціям встановити політику інформаційної безпеки, ідентифікувати ризики, розробити та впровадити відповідні заходи забезпечення безпеки, а також постійно вдосконалювати систему управління інформаційною безпекою[4].

ISO/IEC 27001 є міжнародно визнаним стандартом і використовується в організаціях по всьому світу для забезпечення інформаційної безпеки та

відповідності вимогам законодавства та стандартів. Він допомагає зменшити ризики кібератак, зберегти репутацію та довіру клієнтів, і підтримувати високий рівень інформаційної безпеки в організації[4].

Основні аспекти ISO/IEC 27001 включають:

1. *Захист інформації:* Стандарт встановлює методи та засоби для ідентифікації та управління ризиками для інформації, з метою забезпечення її конфіденційності, цілісності та доступності[4];

2. *Система управління інформаційною безпекою (СУІБ):* ISO/IEC 27001 допомагає створити систему управління, яка враховує вимоги інформаційної безпеки, а також встановлює процедури для оцінки ризиків та прийняття заходів для їх зниження[4];

3. *Постійне вдосконалення:* Стандарт передбачає постійне вдосконалення СУІБ на основі моніторингу та аналізу інформаційних ризиків та інцидентів[4];

4. *Відповідність і сертифікація:* Організації можуть здійснювати самооцінку відповідності до ISO/IEC 27001 або навіть отримати сертифікат підтвердження відповідності від незалежних оцінювальних організацій[4];

5. *Міжнародна акцептованість:* ISO/IEC 27001 - це міжнародний стандарт, що визнається багатьма країнами та галузевими організаціями, тому він допомагає встановити високий стандарт інформаційної безпеки[4].

1.4. Закон України "Про захист персональних даних"[5] є важливим правовим актом, який регулює обробку особистих даних громадян в Україні. Основна мета закону - забезпечити право громадян на приватність та захист їхніх особистих даних[5].

Цей закон визначає права та обов'язки суб'єктів даних (громадян) та тих, хто обробляє ці дані (операторів та обробників). Він передбачає необхідність отримання згоди від суб'єкта даних для обробки їхніх особистих даних та встановлює вимоги до зберігання та захисту цих даних[5].

Закон також передбачає відповідальність для порушення вимог щодо захисту персональних даних і може включати адміністративні штрафи та навіть кримінальну відповідальність для порушників[5].

Забезпечення захисту особистих даних є важливим завданням в епоху цифрового інтернету, і цей закон грає важливу роль у забезпеченні приватності та конфіденційності громадян України[5].

Основні положення Закону про захист персональних даних включають:

1. *Визначення особистих даних:* Закон визначає, що розуміється під особистими даними та установлює категорії таких даних[5];

2. *Права суб'єктів даних:* Закон надає громадянам право на доступ до їхніх особистих даних, вимагання їхньої корекції, анулювання або видалення[5];

3. *Обов'язки операторів та обробників:* Закон встановлює обов'язки операторів (осіб, які визначають цілі обробки даних) та обробників (осіб, які обробляють дані від імені оператора) щодо забезпечення безпеки та конфіденційності особистих даних[5];

4. *Вимоги до зберігання і обробки даних:* Закон визначає вимоги щодо зберігання та обробки особистих даних, включаючи необхідність отримання згоди суб'єкта даних[5];

5. *Відповідальність та штрафи:* Закон передбачає відповідальність за порушення вимог щодо захисту персональних даних, включаючи адміністративні штрафи та кримінальну відповідальність для порушників[5].

Закон України "Про захист персональних даних" вирішує важливі питання, пов'язані з обробкою особистих даних та захистом приватності громадян. Це важливий документ для організацій та осіб, які обробляють особисті дані в Україні. Будь ласка, зверніть увагу на те, що інформація може змінюватися, і слід перевірити актуальний текст закону та інші джерела для отримання оновленої інформації[5].

1.2. Класифікація методів захисту даних

Методи захисту даних включають в себе широкий спектр технологій та стратегій, спрямованих на забезпечення конфіденційності, цілісності та

Рис. 1.1. Класифікація криптографічних алгоритмів

доступності інформації. Один з важливих методів захисту даних - це використання криптографічних технік (рис. 1.1)[6].

Використання цих методів разом із криптографічними техніками допомагає забезпечити високий рівень безпеки для даних та зберегти їх конфіденційність та цілісність[1].

Криптографія використовує різні методи шифрування та розшифрування інформації. Ось деякі основні криптографічні методи захисту даних:

1. Симетричне шифрування: у симетричному шифруванні використовується один і той же ключ для шифрування та розшифрування даних. Популярні алгоритми симетричного шифрування включають DES, AES та інші[1][6];

2. *Асиметричне шифрування:* у цьому випадку використовуються два ключі: один для шифрування та інший для розшифрування. Алгоритми RSA та ECC є прикладами асиметричного шифрування і часто використовуються для захисту даних під час їхньої передачі через мережу[1, 6];

3. *Хеш-функції:* хеш-функції використовуються для створення "відбитків" (хешів) вхідних даних. Це дозволяє перевірити цілісність даних, оскільки навіть невелика зміна вхідних даних призводить до зміни хеш-значення[1];

4. *Цифрові підписи:* цифрові підписи використовуються для підтвердження автентичності документів або повідомлень. Вони створюються за допомогою асиметричного шифрування та служать як підпис відправника[1];

5. *Протоколи обміну ключами:* протоколи, такі як Diffie-Hellman та TLS (Transport Layer Security), дозволяють встановити безпечний обмін ключами для шифрування даних між сторонами[1];

6. *Шифрування в руху та спокою:* для захисту даних під час їх передачі через мережу (в руху) використовуються протоколи шифрування, такі як SSL/TLS. Для зберігання даних (в спокої) на пристроях використовується шифрування даних в пам'яті або на диску[1].

Криптографія грає важливу роль у сфері кібербезпеки та захисту даних. Вона дозволяє забезпечити конфіденційність, цілісність та доступність інформації під час її обробки та передачі, а також попереджає несанкціонований доступ до даних[6].

Існує кілька криптографічних методів шифрування, які використовуються для зберігання даних з метою забезпечення конфіденційності та безпеки[6]. Декілька з найпоширеніших методів включають:

1. AES (Advanced Encryption Standard): це симетричний алгоритм, який використовується для шифрування і розшифрування даних. Зазвичай використовується з ключами довжини 128, 192 або 256 біт[6];

2. RSA (Rivest–Shamir–Adleman): це асиметричний алгоритм, де для шифрування використовується публічний ключ, а для розшифрування - приватний. Часто використовується для обміну ключами та підпису[6];

3. Blowfish: симетричний блочний шифр, який може використовуватися для шифрування даних. Він володіє гнучкістю щодо довжини ключа та є безпечним[6];

4. Twofish: це симетричний блочний шифр, який служить альтернативою AES. Він також є дуже безпечним і володіє хорошою швидкістю[6];

5. Argon2: це алгоритм хешування та процесу затримки, призначений для зберігання паролів. Він ефективно захищає від атак по перебору та атак по словнику[6];

6. bcrypt: інший алгоритм для хешування паролів, який спеціально розроблений для уповільнення атак на перебір[6];

7. SHA-256 та SHA-3: ці алгоритми відносяться до хеш-функцій і використовуються для створення фіксованих довжин хеш-кодів з вихідних даних[6].

Хеш-функції грають важливу роль в технології блокчейн[7]. Вони використовуються для кількох ключових завдань (табл. 1):

- *створення унікального блок-ідентифікатора:* кожен блок у блокчейні має свій унікальний ідентифікатор, який генерується за допомогою хеш-функції. Це дозволяє легко ідентифікувати конкретний блок у ланцюжку:
- *перевірка цілісності даних:* хеш-функції використовуються для створення хеш-кодів транзакцій та блоків. Якщо навіть невеликі зміни внесені до блоку або транзакції, це призведе до істотної зміни хеш-коду. Це дозволяє легко виявити будь-які недостовірності або втручання в дані[6];
- *доказ витрати обчислювальних ресурсів (Proof of Work):* у багатьох блокчейн-протоколах, таких як Bitcoin, хеш-функції використовуються для

завдань "доказу витрати обчислювальних ресурсів". Майнери повинні знайти значення, яке, після хешування, задовольняє певні умови. Цей процес вимагає значних обчислювальних ресурсів і допомагає забезпечити безпеку мережі[6];

- *забезпечення конфіденційності:* хеш-функції можуть використовуватися для шифрування та зберігання конфіденційної інформації, наприклад, в інтелектуальних контрактах на блокчейні[6].

Порівняльна характеристика хешування та шифрування

Таблиця 1.1

Параметр	Хешування	Шифрування
Опис	Хешування використовується для створення фіксованого хеш-коду, який унікально відображає вхідні дані	Шифрування перетворює дані в незрозумілий текст (шифр), який може бути розшифрований з використанням ключа
Розмір вихідних даних	Хеш-коди завжди мають фіксовану довжину, незалежно від розміру вхідних даних	Розмір вихідних даних може варіюватися в залежності від розміру вхідних даних та використаного алгоритму
Обернене перетворення	Немає оберненого перетворення, тобто неможливо отримати вихідні дані з хеш-коду	Шифр може бути розшифрований з використанням ключа для відновлення вихідних даних
Ключі	Не використовуються ключі. Вихідний хеш-код є однаковим для одних і тих самих вхідних даних	Використовуються ключі для шифрування та розшифрування даних

Продовження таблиці 1.1

1	2	3
Призначення	Хешування використовується для перевірки цілісності даних, ідентифікації об'єктів, захисту паролів та багатьох інших завдань	Шифрування використовується для захисту конфіденційності даних, таких як повідомлення, файли та інша конфіденційна інформація

Хеш-функції - це важливий інструмент в області криптографії та інформаційної безпеки. Вони допомагають забезпечити безпеку, цілісність та ефективність операцій в блокчейні, зробивши транзакції і дані надійними та стійкими до змін та атак[7].

Основні аспекти хеш-функцій включають:

- *Унікальність хеш-коду:* Кожному вхідному набору даних відповідає унікальний хеш-код. Навіть найменші зміни в вхідних даних призводять до істотної зміни хеш-коду[7];
- *Фіксована довжина хеш-коду:* Хеш-функції генерують хеш-коди фіксованої довжини, незалежно від розміру вхідних даних[7];
- *Швидкість обчислення:* Хеш-функції спроектовані для швидкого обчислення хеш-кодів, навіть для великих обсягів даних[7];
- *Застосування:* Хеш-функції використовуються для багатьох цілей, включаючи перевірку цілісності даних, підписування повідомлень, ідентифікації користувачів, генерації ключів і багато інших завдань у сфері кібербезпеки та захисту даних[7];
- *Колізії:* Хеш-функції можуть мати колізії, тобто два різних набори даних можуть мати однаковий хеш-код. Однак сучасні криптографічні хеш-функції розроблені з метою ускладнення знаходження колізій[7].

Хеш-функції - це важливий компонент криптографії та інформаційної безпеки. Вони використовуються для перетворення будь-яких вхідних даних в фіксованої довжини вихідні значення (хеш-коди), які намагаються унікально відображати вхідні дані[8]. Ось деталі:

1. *Унікальність ідентифікатора:* Кожному вхідному набору даних відповідає унікальний хеш-код. Навіть невелика зміна вхідних даних призводить до значної зміни в хеш-коді, що робить хеш-функції корисними для перевірки цілісності даних[8];

2. *Фіксована довжина хеш-коду:* Хеш-функції генерують хеш-коди фіксованої довжини, незалежно від розміру вхідних даних. Наприклад, SHA-256 генерує хеш-коди завжди розміром в 256 біт[8];

3. *Швидкість обчислення:* ***Х***еш-функції призначені для швидкого обчислення хеш-кодів навіть для великих обсягів даних[8];

4. *Використання хеш-функцій:* ***Х***еш-функції використовуються для різних цілей, включаючи перевірку цілісності даних, підписування повідомлень, ідентифікації унікальних користувачів, генерації ключів та ідентифікації дублікатів даних[8];

5. *Колізії:* Однією з основних властивостей хеш-функцій є те, що два різних набори даних можуть мати однаковий хеш-код (колізію). Сучасні криптографічні хеш-функції розроблені так, щоб ускладнити знайдення колізій[8];

6. *Приклади хеш-функцій:* Деякі популярні криптографічні хеш-функції включають SHA-256, SHA-3, MD5 та інші. Кожна з них має свої характеристики та застосування[8].

Хеш-функції використовуються в широкому спектрі застосувань, включаючи захист паролів, цифрові підписи, перевірку цілісності файлів, ідентифікацію та безліч інших важливих завдань у сфері кібербезпеки та інформаційної безпеки[8].

1.3. Технологія блокчейн

Найвідоміша і лідируюча з капіталізації криптовалюта Bitcoin принесла із собою технологію розподіленого реєстру – блокчейн. Його можливості за 10 років вийшли далеко за межі операцій з біткоїни та альткоїни. Блокчейн не тільки відкрив нові перспективи, а й став причиною масового божевілля, гострих конфліктів та багатомільйонних афер[1, 9].

Окрім терміна «блокчейн» також часто використовується словосполучення «розподілений реєстр». Насправді з-поміж них існує деяке концептуальне різниця, оскільки розподілений реєстр ширше поняття. Можна навіть сказати, що блокчейн — окремий випадок розподіленого реєстру.

Розподілені реєстри стали в нагоді не тільки для операцій з криптовалютами, а й для створення державних баз даних, систем цифрової ідентифікації, реєстрації прав інтелектуальної власності та бухгалтерського обліку[9].

У межах державних та корпоративних проектів часто створюються розподілені реєстри не з одноранговою, а з ієрархічною структурою, де деякі вузли мають вищий рівень повноважень і здатні впливати на роботу всієї мережі та приймати рішення без підтримки більшості.

Блокчейн спирається на криптографічний протокол і це допомагає[1, 10]:

- з одного боку, вирішити проблему, звану «подвійну витрату» (А дає Б, переконавшись, що вона не дала паралельно С)[1, 10];
- з іншого боку, гарантувати неможливість підробити ідентифікатори заінтересованих сторін[1, 10].

Ілюстрація однієї з реалізацій технології блокчейн зображена на рисунку 1.2. Правові аспекти блокчейну можна розділити на чотири етапи:

- два учасники домовляються про угоду;

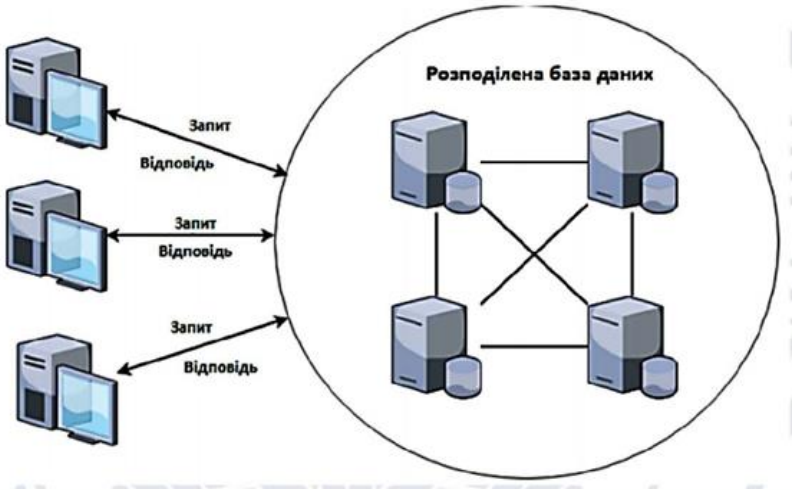


Рис. 1.2. Приклад роботи блокчейна

- за допомогою блокчейна угода шифрується та затверджується на основі підтвердження виконаної роботи;
- далі угода вписується, а потім блокується в останньому блок блокчейна;
- на останньому етапі ланцюжок блоків реплікується на всіх вузлах (учасниках) мережі.

Блокчейн - технологія дуже потужний інструмент, і передбачається, що вона здатна змінити те, як у світі переміщуються кошти, здійснюється захист систем і створюються цифрові ідентифікуючі документи[1].

Необхідно розуміти, як ця технологія працює і які у блокчейна принципові обмеження, оскільки ця технологія дуже скоро буде інтегрована у безліч повсякденних інтерактивних взаємодій (починаючи того, як наймачі платитимуть виконавцям, і закінчуючи тим, як урядові організації забезпечуватимуть цілісність та захищеність своїх комп'ютерних). систем та даних)[1, 11].

У цій технології є набір механізмів, який допомагає системі залишатися незалежною і прозорою. За блоками транзакцій можливо відстежити вірність кожної угоди. Це децентралізована база даних, яка дозволяє виробляти

транзакцію анонімно, миттєво і без участі спеціалізованих посередників. Усі транзакції по рахунку відкриті в блокчейні для будь-кого, і кожна відбивається у вигляді комбінації символів із зазначенням угоди, її суми, одержувача та відправника, а також міток часу[1].

Блокчейн є технологією, що дозволяє проводити транзакції між рівноправними учасниками єдиної мережі (P2P-мережі), при цьому відсутня необхідність стороннього посередника, транзакція здійснюється безпосередньо між учасниками мережі. Інформація про такі транзакції не зберігається в централізованій базі даних. Вона передається на комп'ютери всіх учасників мережі, де дані зберігаються локально[11].

Анонімність користувачів не завжди може бути забезпечена. У літературі зазначено, що «блокчейн у кращому разі напіванонімен, оскільки сам ланцюг можна використовувати математично, щоб побачити ідентифікаційні дані сторін у будь-якій транзакції. Інші критики встановили, що без певного захисту можна прив'язати псевдоніми користувачів до IP-адреси, де було згенеровано транзакцію, з метою відстежити залучені до неї сторони»[1].

Блокчейн представляє собою специфічний цифровий контракт, що дозволяє конкретній особі безпосередньо проводити транзакції з іншою особою та виставляти рахунки. У цьому контексті інформація про транзакції зберігається в комп'ютерній мережі, яка включає комп'ютери як покупця, так і постачальника, що беруть участь у транзакції, а також комп'ютери інших учасників мережі[1].

У цьому випадку банк, як традиційний посередник угод, не є необхідним, оскільки свідками кожної транзакції між постачальником та покупцем виступають інші учасники мережі. Вони можуть підтвердити деталі транзакції, оскільки вся інформація зберігається локально на комп'ютерах усіх учасників.

У світі програмного забезпечення слід відокремити програмне забезпечення з вільним доступом від того, яке захищене законом про інтелектуальну власність[12].

Програмне забезпечення є вільним, лише якщо його ліцензія гарантує чотири основні види свободи:

- свобода використання програмного забезпечення[12];
- свобода копіювання програмного забезпечення[12];
- свобода вивчення програмного забезпечення[12];
- свобода зміни програмного забезпечення та розповсюдження змінених версій[12].

Останні два види свободи можуть застосовуватися тільки в тому випадку, якщо є доступ до вихідного коду, який є в певному сенсі «рецептом» створення цього програмного забезпечення[12].

По суті блокчейн – нова, децентралізована (рис. 1.3), надійна та прозора технологія, яка дозволяє зберігати, обмінювати, перевіряти та верифікувати інформацію, причому ці дії коштують недорого та підтримуються самим користувачем[1].

З цього ми можемо виділити основні правові аспекти технології блокчейн[1]:

1. Централізація чи децентралізація: децентралізація є важливою концепцією, яка пов'язана не лише з блокчейном біткойну. Питання «централізація чи децентралізація» виникає у найрізноманітніших цифрових технологіях. Наприклад, в Інтернеті, що є децентралізованою системою, існує електронна пошта, ядром якої є децентралізована система на основі протоколу відкритого стандарту[1];

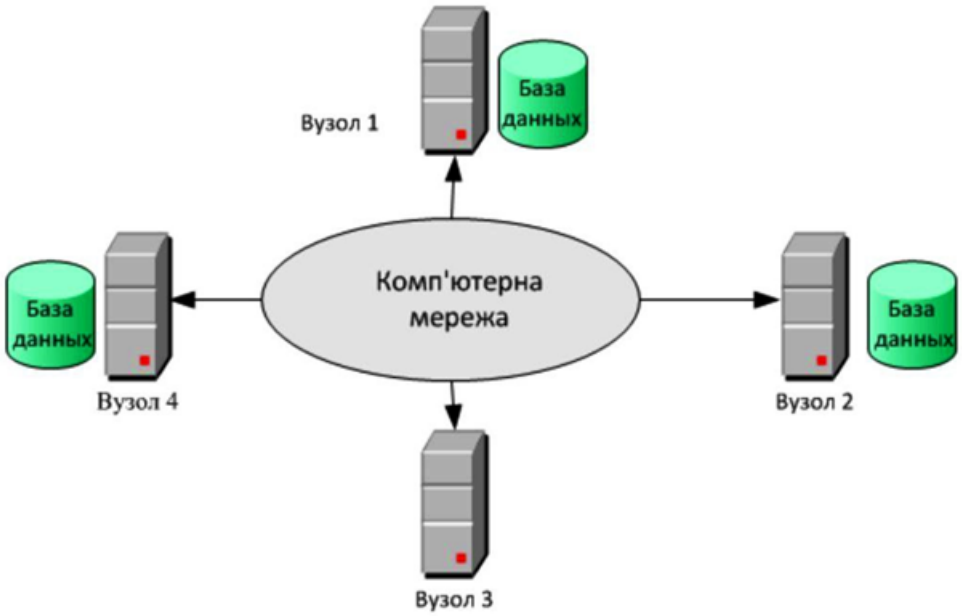


Рис. 1.3. Відмінність централізованої системи

2. *Розподілений консенсус*: основною технічною проблемою, яку доводиться вирішувати при побудові розподіленої системи, незалежно від варіантів її використання є досягнення розподіленого консенсусу. Ця концепція консенсусу характеризує блокчейни і породжує помітні різницю між різними технологіями блокчейна[1];

3. *Відмова від посередників*: принцип відмови від посередників, або видалення третьої довіреної особи, є суттю технології блокчейна. Справді, вона працює без посередників. Прийняття принципу розподілених гросбухів з «позбавленням» фінансового посередництва призводить до того, що угоди, прийняті чи відхилені, стають результатом розподіленого консенсусу, а чи не волевиявлення централізованого установи. Деякі спостерігачі передбачають, що блокчейн із його відмовою від довіреної третьої сторони має «зробити горизонтальною» всю нашу комерційну діяльність[1];

4. *Безпека*: блокчейн є наслідком децентралізованої та розподіленої системи: інформація не зберігається в одному місці, але поширена по мережі[1];

5. *Прозорість та незмінність*: вільний та безкоштовний блокчейн прозорий: він дозволяє отримати доступ до вихідного коду платформи, ознайомитися з інформацією та історією всіх транзакцій чи подій, що відбулися з моменту створення блокчейну. У «блокчейнізованій» системі всі записи незворотні та недоступні для фальсифікації. Іншими словами, коли щось реєструється в цій системі, воно зберігається постійно та доступно для ознайомлення всім учасникам[1];

6. *Простежуваність*: блокчейн – це активний реєстр, що заповнюється в хронологічному порядку, розподілений, перевіряється та захищений від підробки за допомогою розподіленої системи довіри, завдяки якій все, що там зареєстровано, доступне для відстеження та не може бути видалено[1, 13].

Таким чином, ми зможемо застосувати цей принцип простежуваності в різних областях. Таких, наприклад, як продукти, ліки, витвори мистецтва, дорогоцінні метали. Також можливо створити спосіб достовірного відстеження, не вдаючись до неодноразової передачі документів – джерела виникнення помилок та благодатного ґрунту для шахрайства[13].

У блокчейнах по-новому змішано багато старих технологій, які суспільство використало вже тисячі років. Але індустрія блокчейна, насправді, все ще молода, аніж про неї вважають. Наприклад, змішання криптографії та процедури платежу призвело до створення криптовалюти[13].

Але її не треба ототожнювати з криптовалютами. Адже дедалі більше блокчейн-проектів розробляється без внутрішньої фінансової складової[1].

Здійснення платежів з використанням токенів, що лише представляють певну цінність, але не володіють нею безпосередньо, також практикується людством з давніх-давен. Змішання цих двох давно відомих речей призводить до отримання чогось абсолютно нового — криптовалюти. Криптовалюта дозволяє взяти концепцію грошей і перенести її у віртуальне інтерактивне

середовище, зберігши при цьому можливість здійснювати в ній продаж цінності захищеним чином за рахунок використання токена[1].

Також можна сказати, що блокчейн – це велика бухгалтерська книга, або журнал, куди кожен може вносити записи і який кожен може прочитати, розкиданий за величезною кількістю комп'ютерів у всьому світі.

Ця книга активна, складена у хронологічному порядку, розподілена, перевіряється та захищена від фальсифікації за допомогою системи розподілу довіри (консенсусу) між учасниками. Кожен учасник мережі має актуальну копію цього журналу, вміст якого весь час синхронізується з усіма іншими учасниками[1]. Таким чином, блокчейн:

- дозволяє автоматизувати транзакції, не залучаючи у своїй третьої сторони;
- є системою розподіленого консенсусу та довіри;
- інфраструктурою, що забезпечує підтвердження справжності та нотаризацію.

Ідея блокчейна проста, але потужна – вона полягає насамперед у новаторському підході. Являє собою реєстр, що складається з безперервного послідовного ланцюжка блоків з інформацією (рис. 1.4)[1].

Кожен блок містить випадкове число, яке є відповіддю на математичне завдання. Вирішення цієї задачі здійснюється перебором випадкових чисел безліччю комп'ютерів, що використовуються майнерами. Коли завдання вирішено і число збігається, мережа підтверджує рішення і блок приєднується до ланцюжка. Це необхідно для того, щоб унеможливити перебування двох і більше блоків одночасно[1].

Технологія блокчейна включає хешування (перетворення даних будь-якого розміру на короткі значення фіксованої довжини). Хеш – математичне перетворення будь-якої інформації на коротку фразу.

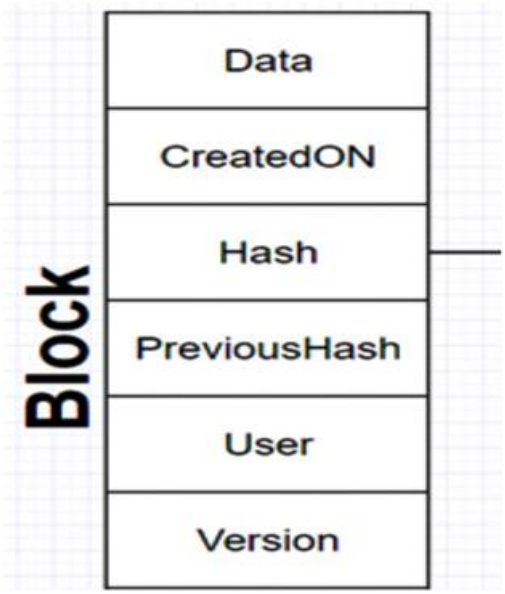


Рис. 1.4. Блок з ланцюга

Наприклад, хешувати слово Delta алгоритмом SHA-256, то отримаємо наступне:

«18833da39fb9b7f8c917fe0220daf9cf12e6524df8fb16e39f04dbe827e2d200»

Сам процес називається хеш-функцією. Хешування широко використовується у криптографії[1].

Криптографія - це мистецтво взаємодіяти захищеним чином на очах у будь-яких третіх сторін[1].

Хешування, своєю чергою, спирається іншу досить давно відому технологію, звану «дерево Меркла» (рис. 1.5). Її суть у тому, що кілька хешей послідовно стискаються до одного хеша, і при цьому зберігається можливість перевірити достовірність кожного елемента даних, який хешувався індивідуально на початку процедури[1].

Щоб максимально уявити блок – станьте блоком. Ви з'явилися на світ завдяки майнерам, які створили вас, вирішивши на своєму потужному обладнанні дуже складне завдання, видане мережею. У вас вкладено унікальне рішення цієї задачі, як у будь-якого іншого блоку[1, 14].

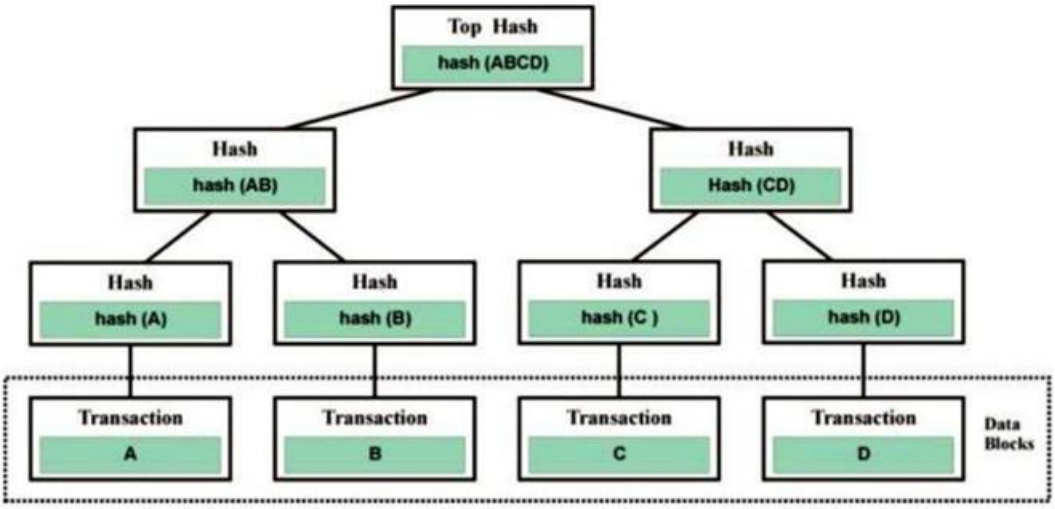


Рис. 1.5. Дерево Меркла

Коли одному з майнерів вдається знайти рішення задачі, мережа підтверджує, що рішення правильне, а майнер, своєю чергою, отримує нагороду у вигляді заданої кількості криптовалюти[1].

Ви міститье інформацію, яка була передана вам у шифрованому вигляді від попереднього блоку. Перед тим, як попередній блок передав вам шифр, ви обмінялися паролями, щоб переконатися, що перед вами є дійсний блок. Тепер ви можете почати записувати транзакції, підтверджуючи їх. Ці транзакції вже були проведені, але були «у повітрі», а як тільки ви стали новим блоком у ланцюжку, вони помістилися всередині вас, тим самим ставши підтвердженими[14]. Але вже на підході нові транзакції, які потребують підтвердження, а вам час передавати естафету новому блоку[1].

Блокчейн є базою даних, що складається з усіх коли-небудь скоєних транзакцій, що у вільному доступі. Ланцюг побудований за певними правилами. Кожен новий блок пов'язаний з попереднім, містить набір записів і додається завжди суворо в кінець ланцюжка (рис. 1.6).

Копії ланцюжків зберігаються і паралельно обробляються відразу на безлічі комп'ютерів, що запобігає можливому збою або втручанню в один із блоків[1].

Якщо спробувати змінити інформацію в блоці на одному комп'ютері, всі інші комп'ютери, що знаходяться всередині мережі, підтвердять, що дана операція недійсна, оскільки зміни відбулися на окремому комп'ютері, а не у всій мережі. Це не позначиться на мережі блокчейн[1, 14].

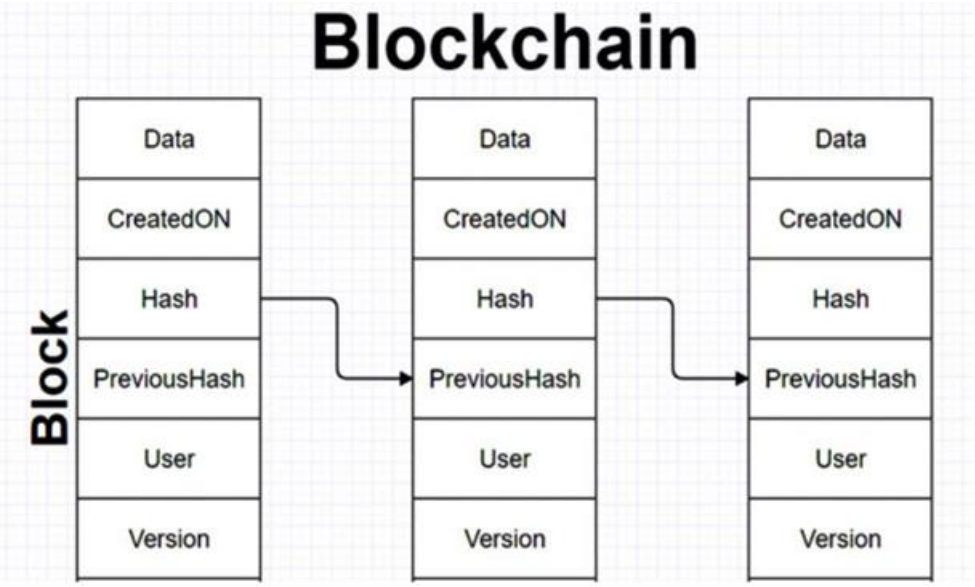


Рис. 1.6. Ланцюг блоків

Будь-який користувач мережі блокчейн має можливість вільного доступу до інформації, що використовується в ній, що робить мережу блокчейн абсолютно прозорою. Мережа блокчейн являє собою ланцюжок з блоків, в яких прописані транзакції з самого початку, з першого блоку. Ви можете у будь-який момент завантажити весь блокчейн до себе на комп'ютер і синхронізувати його в реальному часі, маючи актуальну інформацію про всі транзакції[1, 14].

Йдеться тут не лише про те, як створити найкращу мережу, банк чи забезпечити більш якісне обслуговування. Розвиток блокчейна залежить від того, що роблять люди, і впливають на це не лише його технічні характеристики.

Незважаючи на повну прозорість, також зберігається повна анонімність. Наприклад, здійснюючи транзакцію біткоїну з одного гаманця на інший,

блокчейн залишається лише інформація про суму відправленого біткоїну, адреси гаманців відправника і одержувача, без будь-якої інформації про учасників угоди[1].

1.4. Висновки до розділу

Було розглянуто нормативно-правові аспекти захисту даних в інформаційних системах, проведено класифікацію криптографічних методів, хеш-функції та шифрування. Було проведено їх порівняння.

На основі порівняння криптографічних методів для захисту даних в інформаційних системах було обрано хеш-функцію, як сучасний засіб для підвищення надійності виконання фінансових транзакцій та зберіганню і обробки інформації технологією блокчейн.

РОЗДІЛ 2. АНАЛІЗ МЕТОДІВ ЗАХИСТУ БАЗ ДАНИХ КРИПТОГРАФІЧНИМИ МЕТОДАМИ ТА ТЕХНОЛОГІЄЮ БЛОКЧЕЙН

2.1. Бази даних

База даних (БД) є організованою колекцією даних, яка забезпечує ефективний доступ, управління та оновлення цих даних. Вона є важливою складовою для зберігання та обробки інформації в різноманітних галузях[9]. Давайте детальніше розглянемо основні поняття та характеристики баз даних:

1. *Таблиці та схеми:* у БД дані організовані у вигляді таблиць. Кожна таблиця представляє собою структурований набір даних. Схема бази даних визначає взаємозв'язки між таблицями та структуру даних[9, 15];

2. *Реляційна та нереляційна модель:*

2.1. *Реляційна модель:* Використовує табличну структуру, де дані організовані у вигляді рядків і стовпців. Вона базується на принципі відносин між різними таблицями[15];

2.2. *Нереляційна модель:* Використовує інші формати для зберігання даних, такі як документи, ключі-значення, колонки чи графи. Це дозволяє більшу гнучкість у роботі з різнорідними даними[15];

3. *SQL (Structured Query Language):* SQL використовується для взаємодії з реляційними базами даних. Це стандартний мовний інтерфейс для введення, оновлення та видалення даних, а також для створення та управління схемою бази даних[15].

2.1.1. Реляційні БД

Реляційна база даних - це тип бази даних, який організований за принципами реляційної моделі даних, запропонованої Едгаром Коддом у 1970 році. Основними компонентами реляційної моделі є таблиці, відносини та атрибути[15].

Основні поняття реляційної бази даних:

1. *Таблиці:* База даних складається з таблиць, кожна з яких має унікальне ім'я та складається з рядків і стовпців. Кожен рядок представляє кортеж, а кожний стовпець – атрибут[15];

2. *Відносини (Relationships):* Взаємозв'язки між таблицями визначаються за допомогою ключів. Ключі визначають, як дані в одній таблиці пов'язані з даними в іншій[15];

3. *Атрибути:* Кожен стовпець у таблиці є атрибутом, який описує конкретний характеристики даних[15];

4. *Ключі:* Ключі використовуються для унікальної ідентифікації записів в таблиці. Один з ключів може бути обраний як основний ключ, інші можуть бути використані як зовнішні ключі для визначення відносин між таблицями[15];

5. *SQL (Structured Query Language):* SQL використовується для взаємодії з реляційними базами даних. За допомогою SQL можна створювати, змінювати, видаляти дані та виконувати запити до бази даних[15];

6. *Нормалізація:* Процес нормалізації використовується для оптимізації структури бази даних, зменшення залежності та усунення аномалій даних[15].

До плюсів реляційних баз даних входять стандартизація, простота використання та можливість вираження складних запитів. Однак вони можуть бути менш ефективними в деяких випадках з великим обсягом даних або в випадках, коли структура даних часто змінюється. У таких випадках нереляційні (NoSQL) бази даних можуть бути більш відповідним вибором[15].

Реляційна база даних володіє численними перевагами, які роблять її популярною в різних сферах[15]:

- *Зручність використання:* Реляційна модель даних є логічною та зрозумілою, спрощуючи розробку для програмістів і користувачів[15];

- *Ефективність:* Реляційна база даних забезпечує ефективне зберігання інформації і швидкий доступ до неї за допомогою оптимізованих запитів SQL[15];
- *Надійність:* Цілісність даних гарантується використанням ключів і обмежень, забезпечуючи надійність інформації[15];
- *Масштабованість:* Легко масштабується для обробки великих обсягів даних і великої кількості користувачів[15].

Незважаючи на ці переваги, реляційні бази даних мають деякі недоліки:

- *Обмеження структури:* Вимагають фіксовану структуру, що може бути неактуальною для динамічних або змінюючихся даних[15];
- *Проблеми продуктивності при великих обсягах даних:* При роботі з великими обсягами даних можуть виникати проблеми продуктивності[15];
- *Складність моделювання зв'язків:* Моделювання складних зв'язків між даними може бути складним завданням[15].

Реляційні бази даних широко використовуються в різних сферах індустрії:

- *Корпоративні системи управління базами даних:* Використовуються для зберігання і управління корпоративними даними[15];
- *Веб-додатки:* Застосовуються для зберігання інформації про користувачів, замовлення та інше[15];
- *Системи електронної комерції:* Використовуються для зберігання даних про товари, замовлення і клієнтів у сфері електронної комерції[15].

Нереляційні бази даних стають все більш популярними завдяки своїм гнучким можливостям та здатності ефективно обробляти великий обсяг різноманітних даних. Однак важливо ураховувати їхні особливості та обмеження, аби вибрати оптимальний варіант для конкретного проекту чи застосування[15].

2.1.2. Нереляційні БД

Нереляційні бази даних – це системи управління даними, які не використовують традиційну реляційну модель. Основний принцип нереляційних баз даних – гнучкість у роботі з великим обсягом структурованих та неструктурованих даних[15].

Основні поняття нереляційної бази даних:

1. *Гнучкість у схемі:* NoSQL БД не вимагають строгої фіксованої схеми, що дає можливість зберігати та обробляти різнорідні та змінювані дані[15];

2. *Горизонтальне масштабування:* можливість легко розширювати систему, додаючи нові сервери або вузли, для роботи з великим обсягом даних[15];

3. *Типи NoSQL БД[15]:*

- ✓ *Ключ-Значення (Key-Value):* Redis, DynamoDB;
- ✓ *Документ-Орієнтовані:* MongoDB, CouchDB;
- ✓ *Колоночно-Орієнтовані:* Apache Cassandra, Hbase;
- ✓ *Графові:* Neo4j, OrientDB.

Нереляційна база даних має кілька *переваг*, які роблять її популярною в багатьох сферах:

- Гнучкість у зберіганні даних: можливість зберігати різноманітні типи даних, такі як документи, графи, ключі-значення тощо[15];
- Швидке зберігання та запити: використання спеціалізованих структур для оптимізації швидкості запитів[15];
- Легка масштабованість: можливість легко розширювати БД за рахунок додавання нових серверів (горизонтальна масштабованість) або збільшення ресурсів на існуючому сервері (вертикальна масштабованість)[15];

- Зручність для біг даних (Big Data): ідеально підходить для систем, які опрацьовують великий потік даних (Big Data)[15].

Незважаючи на свої переваги, реляційні бази даних також мають деякі *недоліки:*

- Відсутність стандартізації: відсутність однозначних стандартів призводить до різноманіття нереляційних систем та їхніх особливостей[15];
- Обмежена підтримка запитів: нереляційні БД можуть обмежувати можливості складних запитів порівняно з реляційними[15];
- Складніше управління даними: можливість виникнення проблем з цілісністю та консистентністю даних[15];
- Неуніверсальність: у деяких сценаріях, зокрема тих, де важлива структурованість даних, нереляційні БД можуть бути не найкращим вибором[15].

2.2. Порівняння баз даних з блокчейн

База даних є централізованою бухгалтерською книгою, якою управляє адміністратор. База даних також відрізняється високою якістю, включаючи можливість читати та писати. При цьому лише сторони з відповідним доступом можуть виконувати дії «Запис» та «Читання»[14]. Бази даних також збільшують можливість зберігання кількох копій тих самих даних та їх історії. Це робиться за допомогою надійно централізованого органу, який керує сервером. Централізація дає багато переваг на базі даних. Наприклад, управління базами даних стає простішим завдяки централізації даних. Забезпечення доступу та ефективне зберігання інформації - це не лише зручно, але і швидко. Однак важливо враховувати недоліки. Однією з найсерйозніших проблем є

потенційний ризик пошкодження даних, який вимагає виваженого підходу до захисту інформації[16].

Щоб подолати цей недолік, слід створити кілька резервних копій. Проте це не завжди виконується, оскільки більшість об'єктів довіряють своїм власникам і не використовують резервне копіювання даних. Важливим недоліком є те, що дані можуть бути змінені будь-ким, хто контролює ресурси даних. Це може бути використано через централізовану природу бази даних[15, 16].

Тепер давайте розглянемо джерела даних технічно. База даних використовує структуру для зберігання інформації і може бути опитана за допомогою мови структурованих запитів (SQL). Вона здатна працювати з різними типами даних і підтримувати всі аспекти сучасного підприємства. Базу даних можна легко масштабувати для обробки великої кількості записів[15].

Історія баз даних розпочалася з файлових ієрархічних систем, які мали обмеження і згодом адаптувались до реляційної моделі. Реляційна модель дозволяє працювати з різними базами даних одночасно. Системи управління базами даних організовують дані у таблицях, що складаються з полів або атрибутів[15].

Існують різні типи блокчейну, такі як приватний блокчейн, що працює у закритій екосистемі. Він подібний до бази даних, але зазначається відмінністю у принципах функціонування. Приватний блокчейн успадковує властивості блокчейну, але працює у закритому середовищі, де лише дозволені адміністратором особи можуть брати участь. Одна зі спільних рис – це централізований аспект[15].

Якщо ми порівняємо блокчейн та базу даних, перше, що помітимо, це те, як здійснюється посібник. Блокчейн призначений для децентралізованої роботи, тоді як бази даних централізовані (рис. 2.1). Ця унікальна особливість

блокчейна дає йому важелі, необхідні для того, щоб стати технологією наступного покоління[16].

Децентралізація призводить до істотних змін у реалізації існуючих систем та процесів, що використовуються у різних галузях промисловості. Це дозволяє мережам працювати незалежно та усуває необхідність централізованого управління[15]. З іншого боку, бази даних повністю опираються на централізованому підході[15].

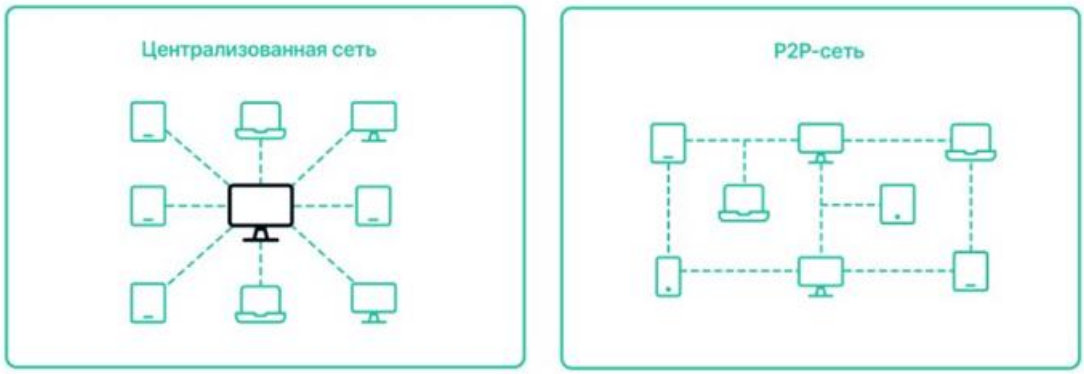


Рис. 2.1. Відмінності централізованої та децентралізованої БД

Жодна традиційна база даних не ґрунтується на принципах децентралізації. Якщо ви маєте на увазі децентралізовану базу даних, то блокчейн входить у цю категорію.

Розгляньмо детальніше, як працює централізація в базах даних. Адміністратор призначається для управління базою даних і має повний контроль над нею. Без адміністратора база даних не працюватиме взагалі. Адміністратор може легко створювати, змінювати та видаляти записи, а також виконувати інші завдання для оптимізації продуктивності. У відношенні до інших користувачів, адміністратор може делегувати ролі, надаючи їм визначені повноваження[15, 16].

Однак ситуація ускладнюється при розгляді різних типів блокчейнів. Базовий блокчейн, представлений у біткойнах, повністю децентралізований, але не є практичним для застосування в підприємствах, які мають справу з

приватними даними та процесами. Саме тому розвивається інший тип блокчейну.

Гібридний блокчейн є найпоширенішим типом блокчейну, який вирішує проблеми приватних організацій, дозволяючи їм повністю керувати налаштуваннями відповідно до своїх потреб. Це відрізняється від приватного блокчейну і бази даних[14].

Архітектурно блокчейн і база даних різні. База даних ґрунтується на клієнт-серверній архітектурі, де клієнт отримує інформацію від централізованого сервера через безпечне з'єднання[16]. З іншого боку, блокчейн використовує мережеву архітектуру розподіленого леджера, де кожен учасник мережі може з'єднатися з іншими за допомогою безпечних криптографічних протоколів. Оскільки централізований вузол відсутній, ноди можуть брати участь у консенсусному алгоритмі, такому як Доказ роботи (Proof-of-Work). База даних, навпаки, вимагає консенсусного алгоритму та залежить від централізованого підходу, де адміністратор має повний контроль. Нижче в таблиці 2 наведено порівняння баз даних та технології блокчейн[15].

Порівняння баз даних та технології блокчейн

Таблиця 2.1

Особливості	**Бази даних**	**Блокчейн**
Структура	Таблиці, рядки, стовпці	Ланцюг блоків, транзакції
Засоби доступу до даних	SQL	API, смарт-контракти
Контроль доступу	Вбудовані ролі та права	Криптографічний доступ
Централізація	Зазвичай централізовані	Децентралізована мережа
Забезпечення	Шифрування, автентикація	Криптографічна безпека

Швидкість та масштаб	Швидкість залежить від розміру та індексації	Розширюється горизонтально, швидкість залежить від конкретної реалізації

Продовження таблиці 2.1

1	2	3
Транзакції	Підтримка транзакцій	Транзакції в кожному блоку
Відміна змін	Зазвичай можливість відміни	В ідеалі неможливість відміни
Довіряючі сторони	Зазвичай потребує довіри	Децентралізована довіра
Змінення структури	Вимагає модифікації схеми	Гнучка структура

Коли мова йде про зберігання та обробку даних, блокчейн та традиційна база даних працюють по-різному. У звичайній базі даних можна легко зберігати та витягувати дані. Для забезпечення правильної роботи програми на початковому етапі використовується CRUD - створення, читання, оновлення та видалення. Це також означає, що дані можуть бути видалені та замінені новими значеннями за необхідності[17].

Блокчейн, навпаки, працює по-іншому у відношенні до зберігання даних. Він підтримує незмінність, що означає, що записані дані не можуть бути видалені або замінені. Це запобігає фальсифікації даних в мережі. Традиційні бази даних не мають незмінності і, таким чином, вони більш схильні до маніпуляцій з боку адміністраторів або хакерів[15, 17].

Блокчейн підтримує лише дві операції: операції читання (для вилучення даних) і операції запису (для додавання інформації). Крім того, блокчейн пропонує прозорість, що дозволяє будь-кому перевірити дані, які були один раз записані в публічному блокчейні[17].

Цілісність блокчейна забезпечується завдяки його незмінності, що означає, що збережені дані не можуть бути пошкоджені або змінені будь-яким чином. Однак вартість реалізації блокчейна вища порівняно з традиційною

базою даних, і він вимагає обдуманої стратегії інтеграції через свою новизну[17].

Традиційна база даних є простою в налаштуванні та масштабуванні, але блокчейн може стати більш економічним рішенням у довгостроковій перспективі, оскільки його пірингові ноди управляють мережею без додаткових витрат на обробку мережі[15].

Те, що неможливо сказати, коли йдеться про привласнення талановитих співробітників, стосується блокчейн технології. Блокчейн є відносно новою галуззю, що також означає, що кількість талановитих фахівців для практичного впровадження блокчейну обмежена. Вартість таких кваліфікованих співробітників блокчейну висока, що може збільшити витрати на впровадження та обслуговування блокчейна на вищому рівні[17].

З іншого боку, знаходження талановитих фахівців, пов'язаних із базами даних, є досить простим завданням. Вони також доступні за доступною ціною, і навіть маленький бізнес може собі дозволити найняти експерта з баз даних.

Щодо швидкості виконання, це є критичним аспектом для порівняння між блокчейном і базами даних. Бази даних відомі своєю високою швидкістю виконання та можливістю обробки мільйонів даних у будь-який момент часу[15, 17].

Блокчейн значно повільніший порівняно з базами даних. Однак це може бути пов'язано з тим, що блокчейн - це відносно нова технологія, і йому все ще потрібен час для розвитку та відповідності стандартам застарілих технологій, таких як бази даних[15].

Коли транзакція виконується в блокчейні, вона проходить всі етапи традиційної бази даних. Проте блокчейн уповільнюється через виконання більшої кількості операцій, таких як перевірка підпису та консенсусні механізми[16].

Централізовані бази даних не стикаються з подібними проблемами, оскільки вони централізовані за своєю природою. Кожна транзакція автоматично перевіряється базою даних і може бути виконана набагато швидше за допомогою черги[15].

Навіть з усім сказаним, блокчейн залишається цікавою технологією для таких використань, як передача вартості, зберігання вартості, грошові операції, довірена перевірка даних, системи голосування та децентралізовані програми. Однак важливо враховувати його обмеження в продуктивності, особливо коли час виконання є важливим фактором. Бази даних залишаються відмінним вибором для критично важливих бізнес-процесів, які потребують ефективності та масштабованості[15, 17].

2.3. Висновки до розділу

Використання технології блокчейн має переваги порівняно з традиційними методами зберігання даних. Блокчейн гарантує безпеку та відсутність централізованих точок доступу, зменшуючи ризик вразливості. Прозорість та можливість перевірки кожної операції додають аудиторську відкритість. Крім того, учасники контролюють свої дані, підсилюючи приватність. Блокчейн також може працювати ефективніше, спрощуючи управління та забезпечуючи швидкодію обробки даних. Ці переваги роблять блокчейн привабливим для захисту та збереження надійних цифрових даних.

РОЗДІЛ 3. ПРОГРАМНИЙ МОДУЛЬ ДЛЯ ЗАХИСТУ БАЗ ДАНИХ У ВЕБЗАСТОСУНКАХ ТЕХНОЛОГІЄЮ БЛОКЧЕЙН

3.1 Модифікації блокчейна

Насправді немає єдиного «офіційного» блокчейну, а є різні типи блокчейнів, які існують незалежно і взаємодіють між собою[1].

Технологія блокчейна може змінювати правила гри: менше централізації, менше влади, більше поділу. Таким чином, блокчейн несе в собі інфраструктуру розподіленої алгоритмічної довіри, або консенсус на вимогу[1].

Таким чином, блокчейн можуть виявлятися специфічні технічні особливості використання його з тими або іншими додатками[1, 18].

Існують різні типи блокчейн-ланцюгів (табл. 3):

1. Публічний блокчейн-ланцюг: Публічні блокчейни є великою розподіленою мережею, у якій запущений власний токен. Приєднатися до неї може будь-хто і на будь-якому рівні. Публічний блокчейн має відкритий код, який підтримується його спільнотою[1]. Внаслідок цього в даному типі блокчейна немає жодного центрального реєстру або довіреної третьої особи. Це найвідоміший тип блокчейна, що лежить в основі цієї технології та відповідає сучасній економіці. Деякі вважають, що при згадці цієї технології слід вживати лише однину – ми, таким чином, говоримо про блокчейн. Його дія заснована на «криптоекономіці», тобто на поєднанні економічних стимулів та механізмів верифікації з використанням криптографії як доказ виконання або доказ участі. Публічний блокчейн за своєю природою повністю децентралізований[1, 19].

2. Ексклюзивний блокчейн-ланцюг: Ексклюзивні блокчейни передбачають наявність центрального органу, що визначає всі ті дії, які стороннім особам дозволено здійснювати у цій мережі. Це також великі

розподілені системи, в яких використається власний токен. Ядро їхнього програмного коду може бути як відкритим, так і закритим[1, 19].

3. *Приватний блокчейн-ланцюг:* Приватні блокчейни зазвичай мають відносно невеликий розмір і зазвичай не припускають використання токена. Їхній членський склад суворо фіксований, всі транзакції відстежуються і контролюються центральним органом. Це той тип блокчейна, який віддають перевагу консорціуму, що мають довірених членів, що маніпулюють конфіденційною інформацією[1, 18].

Типи ланцюгів блокчейну

Таблиця 3.1

Основна мета	**Тип технології блокчейн**
Переміщення цінностей між сторонами, що не є довіреними	Публічний
Торгівля за валюту одного виду	
Забезпечення індентифікації людини	
Запис інформації до відкритого обліку	
Запис інформації про права властності на землю	
Торгівля секьюритизованими активами	Публічний або Ексклюзивний
Створення децентралізованих контрактів	
Запис інформації до закритого обліку	
Забезпечення аудиту записів і систем	
Торгівля цифровими валютами	

Продовження таблиці 3.1

1	2
Торгівля за валюту різного виду	Ексклюзивний
Переміщення цінностей між довіреними сторонами	Приватний

У всіх трьох типах блокчейна використовується криптографія, що дозволяє кожному учаснику будь-якої окремої мережі маніпулювати своїми активами абсолютно безпечним способом, без необхідності мати в ній якийсь центральний орган, уповноважений забезпечувати безпеку її функціонування. Виняток будь-якого центрального органу із структури бази даних — одна з найважливіших і найсильніших сторін технології блокчейну[1, 18].

Технологія блокчейна - це потужний інструмент, оскільки вона дозволяє створювати чесні системи, що забезпечують самокорекцію без необхідності підключення третіх сил для контролю за дотриманням правил. Дотримання встановлених правил блокчейн-технології реалізується з допомогою використання алгоритму консенсусу[1, 19].

У світі блокчейна термін консенсус визначає процес досягнення угоди серед групи тих, хто не довіряє один одному учасників системи. У цьому випадку ними є повні вузли, що функціонують у мережі. Повні вузли перевіряють транзакції, що були створені в мережі та мають бути записані до реєстру[1, 20].

Виділимо основні принципи консенсуса *публічного* блокчейна:

- ✓ характеристики: громадська мережа без посередників та без цензури;
- ✓ консенсус (proof of work): дорогий, «повільний», із властивою йому компенсацією мережі.

Виділимо основні принципи консенсуса *приватного* блокчейна:

✓ особливості: приватний або одержуваний (різні права доступу до платформи), учасники відомі чи ідентифіковані, сектор регульований;

✓ консенсус: між відомими учасниками функціонування зовнішнє стосовно платформи (відповідальність беруть він один чи кілька уповноважених представників приватного блокчейна)[1, 21].

На рис. 3.1 зображено досягнення консенсусу в технології блокчейн. У кожній конкретній реалізації блокчейн-технології використається власний алгоритм досягнення угоди в мережі при додаванні нової транзакції. Існує безліч різних моделей досягнення консенсусу, оскільки в кожній реалізації блокчейна використовуються різні типи вхідних записів. В одних ланцюгах блокчейна зберігаються валютні активи, в інших — деякі дані, а треті є захисними системами для контрактів[1, 20].

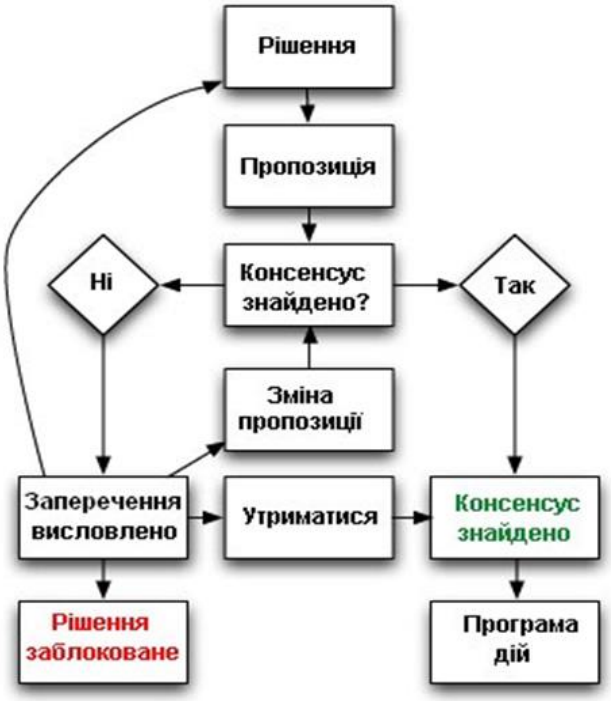


Рис. 3.1. Блок-схема досягнення консенсусу

У системі біткойна, наприклад, між членами мережі здійснюється торгівля з урахуванням її токена[22]. Цей токен має певну ринкову вартість,

тому вимоги щодо продуктивності, масштабування, цілісності, методів захисту від зовнішніх загроз та відмов обладнання мають бути дуже високими. Робота системи біткойна побудована у припущенні, що зловмисник може спробувати порушити історію записів про транзакції метою викрадення токенів. У біткойне ця небезпека запобігає за рахунок використання моделі консенсусу, званої «доказ виконання роботи»[1].

Такий підхід дозволяє успішно вирішити завдання візантійських генералів: «Як можна дізнатися, що інформація, що надійшла, не була спотворена зсередини або ззовні?»[23]. Оскільки зміна даних або маніпуляція ними можлива практично завжди, забезпечення надійності даних це велика проблема в комп'ютерних науках[1].

Для більшого розуміння розглянемо проблему Візантійських генералів (рис. 3.2). Проблема, чи теорія, візантійських генералів – це математична метафора, у якій розглядається проблема перегляду безвідмовності засобів зв'язку та цілісності співрозмовників[1, 21].

Ось як виглядає проблема: генерали, кожний з яких командує окремою армією, мають координувати свої дії, щоб укласти в облогу місто. Генерали спілкуються за допомогою надійних кур'єрів, але деякі з генералів виявилися зрадниками і прагнуть зірвати план нападу (візантійська помилка, таким чином, являє собою збій, що полягає в наданні недостовірної або суперечливої інформації). Таким чином, напад може зірватися, якщо генерали не дійдуть консенсусу[1, 22].

Потрібно переконатися, що лояльні генерали зможуть все ж таки домовитися і погодити план битви. Слід координувати довірчі відносини за допомогою повідомлень, написаних і підписаних (без можливості підробки), які генерали передають один одному, поділяючись намірами з усіма генералами[1, 20].

Таким чином, ми повертаємось до консенсусу «proof of work». Технологія блокчейну надає перше і, можливо, єдине вирішення проблеми візантійських

генералів. Ймовірно, вперше в історії людства вдається створити та зберегти реєстр, відкритий для широкої публіки та досить безпечний для кожного[1, 20].

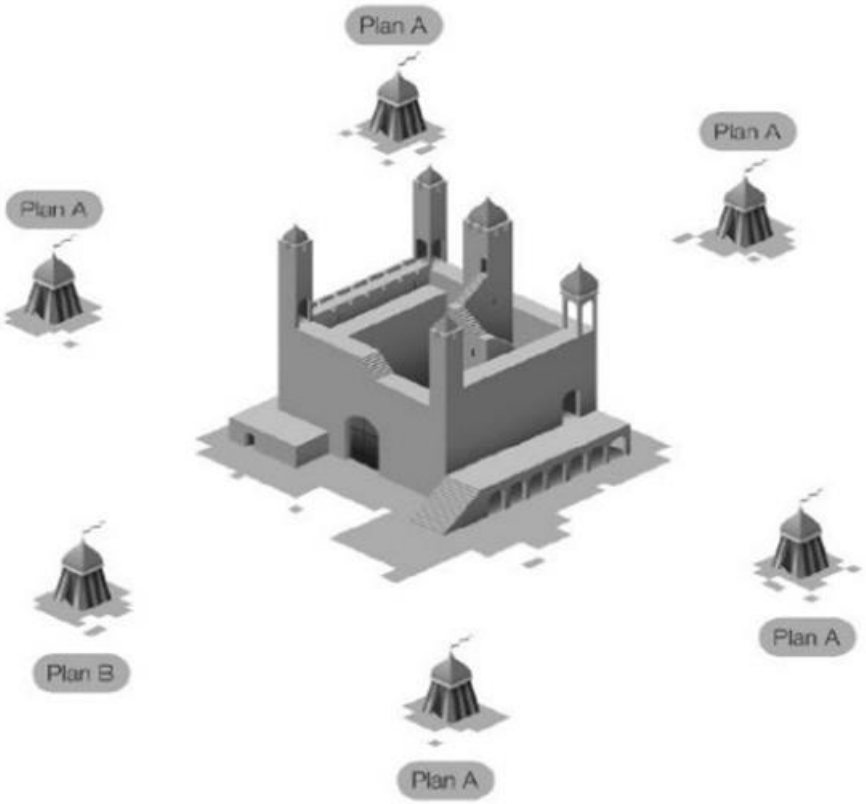


Рис. 3.2. Проблематика Візантійських генералів

Стійкість до візантійських збоїв - це здатність системи продовжувати функціонувати, часом у скороченому обсязі, не виходячи повністю з ладу, коли частина її компонентів працює неправильно[20]. Історично цю систему відмови стійкості розробили військові під час холодної війни для забезпечення безперервної роботи складної замкнутої мережі[1, 20].

3.2. Аналіз та різновид сфер використання технології блокчейн

В останні роки відзначається великий галас навколо технології блокчейну та криптовалют, які функціонують на її основі[26]. Частина цього галасу відбувається просто через коливання ринкової вартості криптовалют і страху,

що блокчейн-технологія зруйнує багато господарських та державних інститутів. Як би там не було, величезні суми грошей було витрачено на дослідження та нові розробки, оскільки власники токенів не хочуть відстати від часу, а підприємці прагнуть досліджувати нові бізнес-моделі[1, 23].

Коли заходить мова про пошук сприятливих можливостей для впровадження блокчейн-технології з метою підвищити цінність організації, часто виникає питання «Де саме технологія блокчейна може виявитися корисною і чим вона відрізняється від існуючих технологій?»[23].

Блокчейн-ланцюги є особливий тип баз даних, тому їх можна використовувати у всіх випадках, у яких можуть використовуватися будь-які інші види баз даних. Однак може виявитися і так, що немає жодного сенсу заново долати труднощі і нести додаткові витрати там, де звичайна база даних і так чудово справляється з поставленим завданням[23].

Але якщо вам доводиться ділитися важливою інформацією з іншими сторонами, яким ви не цілком довіряєте, або якщо ваші дані потребують аудиту, або існує ризик, що дані будуть порушені або підмінені зсередини або ззовні, ви дійсно зможете оцінити всі переваги використання блокчейна певного типу. На жаль, на жодне з поставлених вище питань не можна дати простих відповідей, і прийняти правильне рішення може виявитися непросто[23].

В даний час ми знаємо всього про кілька способів його використання, але, схоже, можливостей застосування блокчейна дуже багато, причому в різних галузях економіки і суспільства. І ці можливості будуть множитися з появою дедалі нових технологій блокчейна[24].

Метою створення блокчейну біткойн у 2009 році було не служіння світові фінансів, а навпаки – його заміна. З часом банки почали усвідомлювати той факт, що ця технологія може порушити їхню бізнес-модель і є одночасно і загрозою, і можливістю[1, 24]. Ось короткий список областей використання технології блокчейну[25]:

✓ фінанси (платежі, що проходять миттєво та практично безкоштовно між двома сторонами);

✓ страхування (мікроконтракти, мікроплатежі, групове страхування, ефективніше управління ідентифікацією клієнтів та пов'язаних з ними даних, сертифікація походження товару);

✓ держава (прозора та безпечна система голосування, збір податків, кадастри);

✓ електронна комерція (прості та безпечні платежі в Інтернеті);

✓ інтернет речей;

✓ промисловість (управління підключеними об'єктами та автономізація об'єктів для укладання угод);

✓ ідентифікація відбитків пальців;

✓ логістика (управління процесами та контрактами за допомогою алгоритмічних процесів);

✓ харчування (відстеження інформації, що відноситься до партії товарів, від збору до упаковки);

✓ інтелектуальна власність (статті, фотографії, музика, ілюстрації);

✓ угоди з нерухомістю у країнах, де немає земельного кадастру;

✓ аутентифікація творів, предметів, цінностей;

✓ навчання (перевірка справжності дипломів);

✓ охорона здоров'я (відстеження медикаментів, забезпечення безпеки медичних даних, управління даними пацієнтів);

✓ енергетика (розумні мережі, розумні будинки, розумні міста)[25].

3.2.1 Використання технології блокчейн зі смарт-контрактами

Смарт-контракт - це віртуальний еквівалент звичайних договорів[26], програма, що автоматизує виконання дій при виконанні сторонами угоди

певних умов. Наприклад, вона може автоматично відправляти гроші продавцю, якщо товар відповідає узгодженим стандартам. Ця технологія дозволяє безпечно обмінюватися криптовалютами, грошима, цінними паперами та іншими благами та послугами безпосередньо між учасниками угоди, із уникненням посередників[1, 26].

Для кращого розуміння розглянемо смарт-контракти на прикладі покупки побутової техніки, такої як ноутбук. Наприклад, якщо ви хочете придбати ноутбук, ви можете шукати найнижчу ціну в Інтернеті та натрапити на маловідомий магазин. Щоб уникнути великих передплат, ви не хочете відправляти всю суму передоплати, особливо якщо ви раніше не чули про цього продавця. З іншого боку, магазин може не згодитися вислати товар, бо ви можете передумати, і тоді продавець повинен буде покрити витрати на транспортування товару до вас і назад, що вплине на його прибутковість[1, 27].

В таких випадках смарт-контракти можуть бути корисними. Наприклад, існує програма, яка дозволяє[27]:

- обрати ноутбук у Інтернет-магазині;
- перерахувати суму для оплати на спеціальний рахунок, а не на рахунок продавця;
- повідомити продавця про отримання оплати на спеціальний рахунок;
- отримати придбаний товар поштою;
- повідомити продавця про отримання товару, після чого сума покупки зараховується на рахунок магазину[1, 27].

Це набагато зручніше, ефективніше і швидше, ніж оплата всіх витрат передплати продавцеві, а потім повна вартість товару під час отримання[27].

Однак, на практиці впровадження даного прикладу стає вельми складним завданням, оскільки смарт-контракт повинен взаємодіяти як із веб-сайтом Інтернет-магазину, так із платіжною системою та поштовою службою, що несе технологічні та юридичні ризики._Таким чином, на сьогоднішній день смарт-

контракти в основному використовуються для операцій, пов'язаних із купівлею чи продажем криптовалют, де сторони угоди мають відкриті рахунки на одній біржі чи в блокчейн-платформі[1, 27].

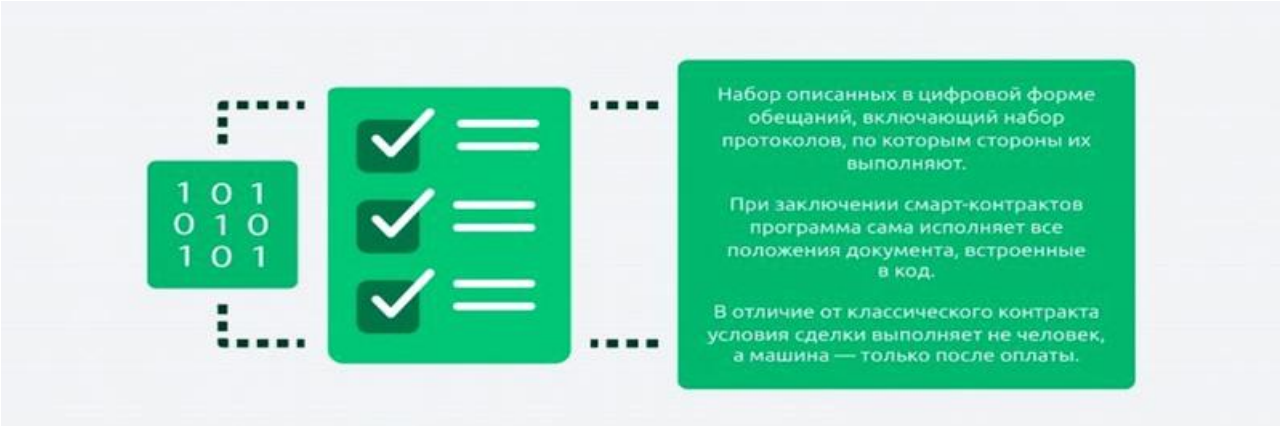


Рис. 3.3. Створення смарт-контракту

Ключові компоненти смарт-контракту включають учасників угоди з цифровим підписом, які узгоджують або відхиляють відповідність товару чи послуги визначеним раніше вимогам, предмет договору (товар чи послуги, що обмінюються на грошові кошти), умови, які автоматично спричиняють обмін благами (наприклад, відповідність стандартам якості) та децентралізовану платформу, на якій розміщений алгоритм (програмний код) самого смарт-контракту. Такі контракти можуть бути повністю автоматизованими, містити переважно паперовий носій з частковим перенесенням пунктів угоди в смарт-контракт (наприклад, проведення транзакцій), або мати копію на паперовому носії[28].

Порівнюючи смарт-контракти зі стандартними договорами, які застосовуються повсюдно, важливими аспектами виявляються носії інформації (комп'ютерний алгоритм для смарт-контракту та папір для стандартного договору), основа документа (програмний код для смарт-контракту та правові норми для стандартного договору), можливість зміни умов (неможливість для смарт-контракту та можливість для стандартного договору), складність складання (висока для смарт-контракту та середня для стандартного договору),

виконання умов (автоматичне для смарт-контракту та залежне від сторін для стандартного договору), застосування покарань (автоматичне для смарт-контракту та через суд для стандартного договору), наявність посередників (без посередників для смарт-контракту та їх можлива участь для стандартного договору), валюта розрахунків (криптовалюта для смарт-контракту та реальні гроші для стандартного договору), час проведення операції (миттєво для смарт-контракту та залежно від часу для стандартного договору) та місцезнаходження сторін (можливість віддаленого підписання для смарт-контракту та часто необхідна особиста зустріч для стандартного договору)[28].

Ризик шахрайських операцій практично виключений для смарт-контрактів, в той час як він є невеликим для стандартних договорів. Зробивши порівняння, можна виділити переваги та недоліки смарт-контрактів. Серед переваг - економія часу та ресурсів, менші витрати без посередників, додаткова безпека від застосування технології блокчейн і швидка перевірка умов виконання контракту. Проте є також і недоліки, такі як можливі помилки та вразливості в програмному коді смарт-контракту, складність в його побудові, ризик втрати ключів доступу та паролів, систематичне врахування умов контракту без урахування форс-мажорних обставин та відсутність законодавчої бази для використання смарт-контрактів[29].

При більш широкому впровадженні технологій блокчейн, а також синхронізації таких платформ з іншими програмами, що використовуються в повсякденному житті, і врегулюванні смарт-контрактів на законодавчому рівні, вони можуть стати широко застосованими у різних галузях, включаючи облік та передачу прав власності, операції з цінними паперами, міжнародні розрахунки, ідентифікацію особистості, фінансову звітність, обробку платежів по кредитах, передачу активів за заповітом, перевірку відповідності товарів стандартам, зберігання медичних даних та передачу інших цифрових активів[29].

Смарт-контракти серед Ефіріуму схожі зі звичайними договірними угодами крім те, що немає центральної сторони, що забезпечує їх виконання. Протокол Ефіріуму "примушує" до виконання смарт-контрактів, чинячи економічний тиск[28]. Також гарантується дотримання всіх вимог, якщо вони були вказані в блокчейні Ефіріуму, оскільки ця система може перевіряти, чи були дотримані певні умови. Якщо деякі умови не зафіксовані в Ефіріумі, гарантувати його дотримання буде набагато важче[1, 28].

Розглянемо криптовалюту Ефириум. Проект Ефіріум — один із найрозвиненіших і найдоступніших блокчейнів у цій екосистемі. Він також є абсолютним лідером у блокчейн-інноваціях та способах їх використання. Розуміти особливості цієї технології дуже важливо, оскільки Ефіріум лідирує в підтримці смарт-контрактів і децентралізованих організацій[1, 28].

Можливо, один із найскладніших блокчейнів із числа коли-небудь створених. Він має власну повнофункціональну мову програмування, що дозволяє розробникам створювати додатки будь-якого типу. Протокол Ефіріуму може виконувати практично все, що можливо здійснити засобами будь-якої звичайної мови програмування, і це крім того, що він вбудований всередину блокчейна, що дає йому додаткові переваги і рівень захищеності. Якщо ми здатні уявити собі якийсь програмний проект, то він може бути реалізований серед Ефіріуму[1, 28].

Повна за Т'юрінгом мова програмування — це головна особливість, яка робить блокчейн Ефіріуму набагато потужнішим (порівняно з Біткойном) інструментом створення нових програм[29]. Мова сценаріїв середовища Ефіріуму дозволяє створити додаток, подібний до Твіттера, за допомогою лише декількох рядків коду, забезпечуючи йому при цьому виняткову захищеність[1, 29].

Найбільш революційним та викликаючим дискусії нововведенням Ефіріуму є самоврядні та децентралізовані додатки. Вони можуть керувати

такими об'єктами, як цифрові активи та децентралізовані автономні організації[1, 29].

Децентралізовані програми були створені замість централізованого управління активами та організаціями. Ця структура для багатьох виявилася дуже привабливою, оскільки давно вже відомо, що абсолютна влада розбещує абсолютно. Для тих, хто боїться втратити контроль, використання структур цього матиме серйозні наслідки. Звідси можемо виділити основні принципи роботи децентралізованих додатків[15]:

1. Група людей створює смарт-конракт для створення організації та управління нею;
2. Учасники виділяють організації кошти і отримують замість токени, які мають їх власність[1, 15].

Ця схема працює подібно до акцій компаній, але в даному випадку учасники отримують контроль над фондами з першого дня. Коли необхідну суму зібрано, організація починає функціонувати та її члени можуть почати подавати на розгляд пропозиції про те, як і на що витрачати наявні кошти[28]:

✓ за кожною пропозицією члени організації проводять голосування;

✓ для кожної пропозиції, коли заздалегідь встановлений для нього час голосування закінчився і було зібрано певну кількість голосів, воно або приймається, або відхиляється;

✓ певні члени виступають як підрядники для обслуговування організації[1, 28].

На відміну від більшості традиційних інвестиційних інструментів, де рішення про інвестування приймає лише центральна ланка, члени децентралізованої автономної організації контролюють усі 100% активів. Щодо всіх інвестицій та інших рішень завжди проводиться голосування. Така організаційна структура загрожує повною відмовою інституту традиційних фінансових менеджерів[1, 29].

Розподілені автономні організації будуються за допомогою програмних інструментів і їх код не може змінюватися в процесі функціонування[30]. Привабливість такого підходу полягає в тому, що зловмисні хакери не зможуть маніпулювати фондами у традиційному сенсі. Проте хакери все ж таки можуть знайти способи змусити програмне забезпечення функціонувати непередбачено і тим самим спробувати вивести кошти. Незмінна природа програмного коду розподілених організацій робить практично неможливим виправлення виявлених у ньому помилок, коли ці організації вже функціонують серед блокчейна Ефіріуму[1, 28].

3.2.2 Використання технології блокчейну з фінансовими операціями

Блокчейн біткойна найбільш явно демонструє всі аспекти блокчейн технології. Це та основа, з якою порівнюють всі інші блокчейн проекти, і та структура, в якій були розроблені практично всі концепції нової технології (рис. 3.4)[1].

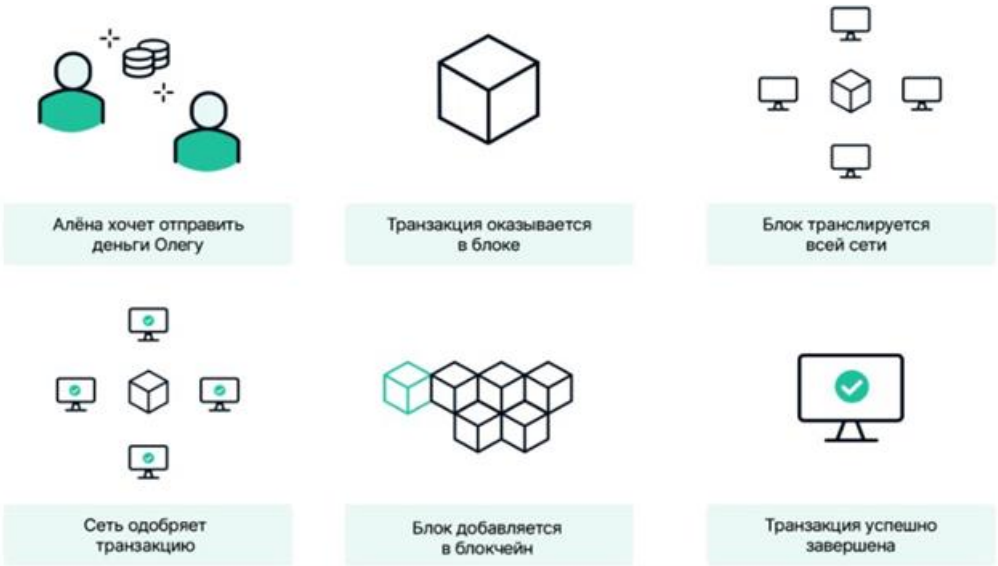


Рис. 3.4. Технологія блокчейн в криптовалюті біткоїн

Біткойн - одна з найбільших і найпотужніша блокчейн-мережа у світі. Спочатку вона була створена для підтримки криптовалюти з тією самою назвою. Тому цілком природно, що для створення в блокчейні Біткойна нового запису необхідно відправити якусь кількість біткойнів з одного рахунку на

інший. Це однорангова мережа: учасники утворюють однорангову мережу, спілкуючись через Інтернет[1, 30].

Коли ви пересилаєте біткойни з одного рахунку на інший, відомості про цю транзакцію записуються в блокчейн Біткойна. Після того як транзакція внесена до блокчейну, цю інформацію вже неможливо звідти видалити — відомості про вашу операцію зберігатимуться в блокчейні Біткойна стільки, скільки він існуватиме. Ця концепція незмінності є надзвичайно потужною - це найважливіша якість будь-якого блокчейн-ланцюга[1].

Існує кілька способів додати в запис транзакції трохи додаткової інформації, але в деяких випадках ці методи необов'язково дозволяють отримати повідомлення, що легко читається. У цьому розділі буде розказано, як можна вбудувати повідомлення безпосередньо в біткойн-транзакцію[1, 29].

Коли новий комп'ютер намагається підключитися до мережі, його першим завданням є знайти інші підключені до неї комп’ютери. Після того, як комп'ютер буде підключений, другий крок - завантажити базу даних всіх операцій, що здійснювалися з моменту запуску проекту, транзакцій, що полягали в передачі певної кількості біткойнів з одного рахунку на інший[1].

Обліковий запис ідентифікується біткойн-адресою, яка схематично аналогічна номеру рахунку в банку. Щоб стати законною, кожна транзакція має бути підписана (у криптографічному сенсі цього терміна) за допомогою асиметричного шифрування або шифрування з подвійним ключем (відкритим та закритим)[30].

На вході транзакція отримує посилання на попередню транзакцію, яка підтверджує той факт, що згадані в угоді кошти реальні, а на виході вона виробляє один або кілька биток адрес з відповідними приписаними до них сумами. Входи та виходи будь-якої транзакції завжди збалансовані. Тим не менш, ця нова транзакція не відразу визначається як допустима, тому що вона повинна спочатку бути включена до блоків, що складається з набору блоків транзакцій[30].

Впровадження даних у біткойн-адресу гарантує, що вони будуть легко читаються. Це можна зробити, використовуючи “красивий” біткойн-адресу. Красивий біткойн-адресу можна уявляти як гарний номерний знак на автомобілі. Шестилітерну красиву біткойн-адресу можна отримати безкоштовно, тоді як за довші вам доведеться заплатити. Чим довша красива адреса, тим дорожча вона коштує[1].

А як ми вже знаємо, що блокчейн – це структура даних у вигляді «ланцюжка блоків», але це зв'язування в ланцюг насправді лише частину розподіленого протоколу реєстру. Отже, у ширшому сенсі логічно назватиме ці технологічні платформи, ці блокчейни, «розподіленими протоколами реєстру» (з відкритим або закритим реєстром)[30].

Ключовий аспект відкритого розподіленого реєстру – це характер розподілу даних та ефективність алгоритму консенсусу, який визначає істинність транзакцій, зареєстрованих у різних вузлах мережі. Саме на підставі цього алгоритму було виведено більшість властивостей розподіленого реєстру[1, 15].

Блокчейн часто критикують за анонімність. Розглядаючи питання під більш технічним кутом, розумієш, однак, що побоювання можуть бути значною мірою зняті: біткойн не настільки анонімний, як здається… З одного боку, якщо ми не повинні забезпечувати підпис інформації для створення портфоліо чи надсилання транзакції, то з іншого боку, все, що відбувається в ланцюжку блоків біткойна, прозоро, що дозволяє публічно відстежувати всі угоди. Таким чином, кожен може створити провідник, який відстежує дані блокчейна[1, 28].

У мережі біткойн особистість користувача прихована за криптографічним псевдонімом, який може бути змінено за бажанням власника. Транзакції підписуються псевдонімом і поширюються на загальнодоступну мережу для перевірки їх справжності та призначення біткойнів новому власнику. Майже всі криптовалюти, що існують, використовують прозорий блокчейн, і тільки мала

частина проектів намагається зробити його непрозорим. Ось компоненти розподіленого протоколу консенсусу[29]:

- ✓ жетони (наприклад, криптографічна валюта, така як біткойн);
- ✓ механізм консенсусу (наприклад, «proof of work», або підтвердження виконання роботи);
- ✓ структура (наприклад, блокчейн);
- ✓ мережу учасників (вузлів);
- ✓ набір правил (наприклад, протокол Ripple)[1].

Протокол Monero використовує метод одноразового кільцевого підпису, дуже потужну технологію досягнення анонімності, що дозволяє повністю приховати транзакцію[1, 30].

При формуванні транзакції випадково вибирається ланцюжок інших блоків, які підписуються по кільцю, щоб створити цифровий відбиток пальців для публікації. Цей відбиток, що характеризує транзакцію, називається key-image. Ця хитрість ховає від спостерігачів справжній підпис, гарантуючи, що угода, безсумнівно, є законною і що вона не є шахрайством[1, 30].

За допомогою ключів view-keys також не можна ідентифікувати одержувача платежу. Транзакція надсилається не за допомогою відкритого ключа, але на адресу, яка буде використана лише один раз. Тільки одержувач, який має у своєму розпорядженні правильний view-key, має право на читання транзакції, яка йому призначена[1].

3.3. Опис реалізації програмного модуля

Програмний модуль реалізує захищене сховище на основі технології блокчейн. Рішення має забезпечувати зберігання інформації без можливості

модифікації (крім повного стирання всіх даних), а також, має дозволяти зберігати довільний набір даних як рядки розміром до 5 МБ.

Програмна реалізація створення захищених баз даних представлено на рисунку 3.5 (див. Додаток А).

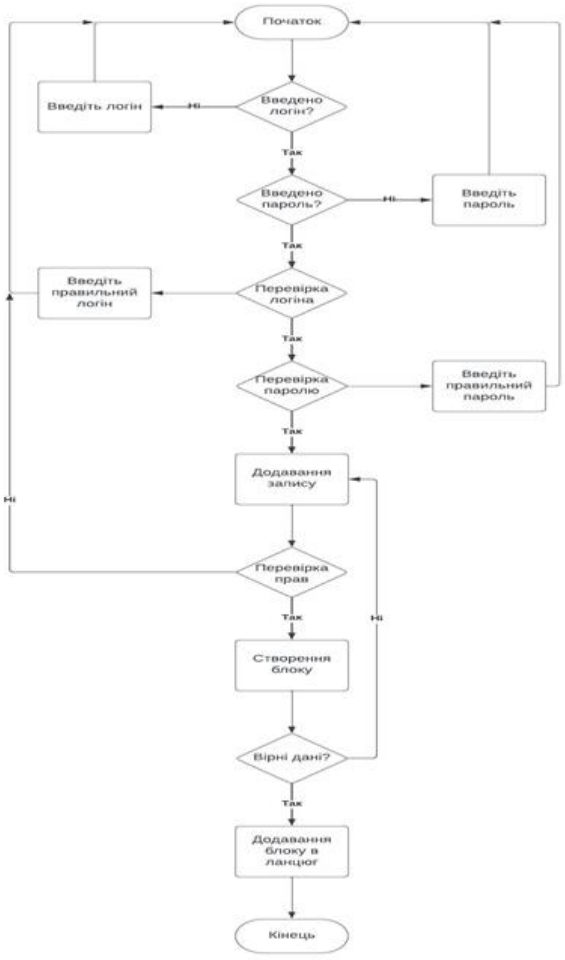


Рис. 3.5. Алгоритм розробленого програмного модуля

Розглянемо код даного вебзастосунку. В даному фрагменті коду створюється Генезис блок. Саме з нього починається будь-яка база даних блокчейн.

```
public Block()
    {
        Id = 1;
        Data = "Genesis block";
        CreatedOn = DateTime.UtcNow;
        PreviousHash = "Underfined";
        Role = "Underfined";
        Hash = GetHashUseSHA256(GetConcatedString());
    }
```

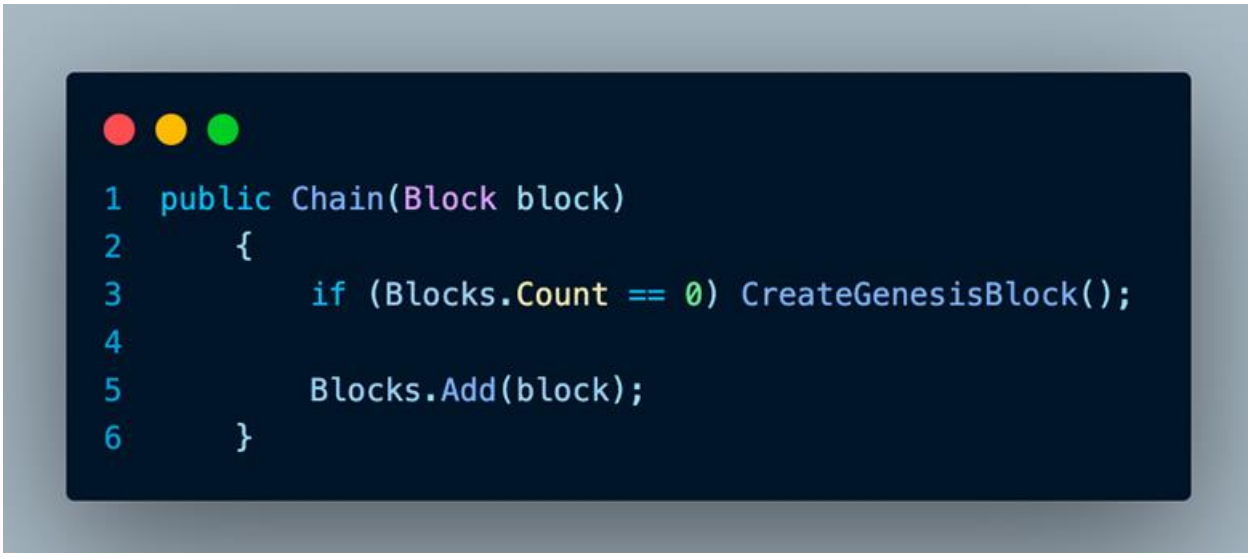
Рис. 3.6. Генезис блок

Тепер хешуємо всі дані нашого блоку. Для цього використовуємо технологію хешування SHA-256.

```
private string GetHashUseSHA256(string concatedString)
    {
        using SHA256 sha256 = SHA256.Create();
        StringBuilder hash = new StringBuilder();

        var message = Encoding.ASCII.GetBytes(concatedString);
        var hashValue = sha256.ComputeHash(message);

        foreach (byte x in hashValue)
            hash.Append(string.Format("{0:x2}", x));

        return hash.ToString();
    }
```

Рис. 3.7. Хешування

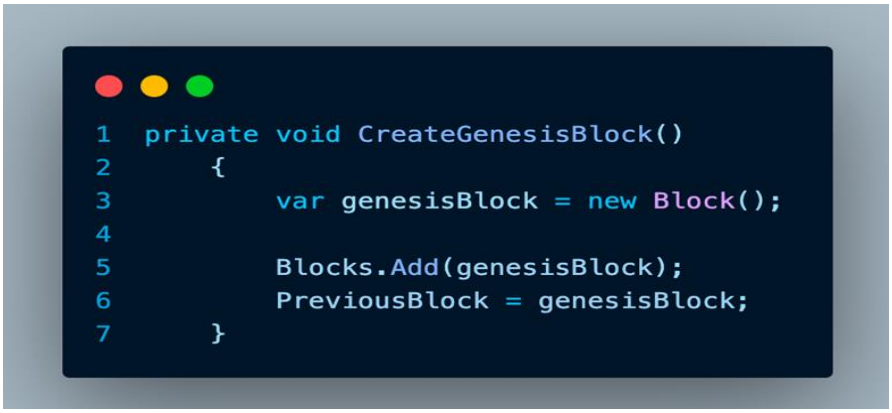
При додаванні в ланцюг перевіряємо базу даних. У разі відсутності записів, створюємо новий ланцюг.



```
public Chain(Block block)
    {
        if (Blocks.Count == 0) CreateGenesisBlock();

        Blocks.Add(block);
    }
```

Рис. 3.8. Додавання в ланцюг

Продивимось функцію створення ланцюга блоків. Вона викликається, якщо при додаванні запису був відсутній ланцюг.



```
private void CreateGenesisBlock()
    {
        var genesisBlock = new Block();

        Blocks.Add(genesisBlock);
        PreviousBlock = genesisBlock;
    }
```

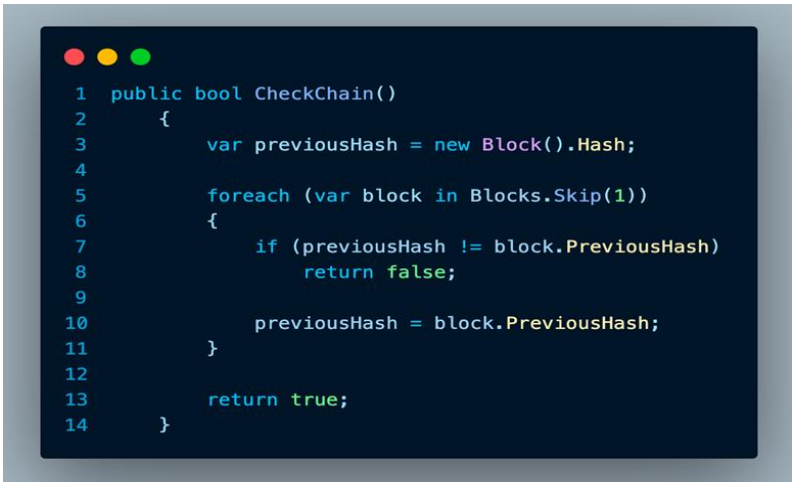
Рис. 3.9. Створення ланцюга

Якщо ж записи вже були, то викликаємо функцію, додавання запису до вже існуючого ланцюга.

```
public void Add(string data, string role)
    {
        var block = new Block(data, role, PreviousBlock);

        Blocks.Add(block);
        PreviousBlock = block;
    }
```

Рис. 3.10. Додавання блоку

Дана функція створена для перевірки ланцюга записів на правильну послідовність.



```
public bool CheckChain()
    {
        var previousHash = new Block().Hash;

        foreach (var block in Blocks.Skip(1))
        {
            if (previousHash != block.PreviousHash)
                return false;

            previousHash = block.PreviousHash;
        }

        return true;
    }
```

Рис. 3.11. Перевірка ланцюга

В даному методі за допомогою атрибуту Authorize вказуємо, що додавати записи можуть тільки авторизовані користувачі.

```
[Authorize(Roles = "admin, user")]
async Task<IResult> PostApiDataHandler(HttpContext context, Chain chain)
{
    var form = context.Request.Form;
    chain.Add(form["data"], form["role"]);

    return Results.LocalRedirect("/");
}
```

Рис. 3.12. Додавання записів користувачами

При введені записів користувачем при аутентифікація, створюємо куку на основі його даних та шифруємо її. Далі дана купа буде використовуватись для авторизації цього користувача для перевірки його прав.

```
async Task<IResult> PostSignInHandler(string? returnUrl, HttpContext context)
{
    var form = context.Request.Form;

    if (!form.ContainsKey("login") && !form.ContainsKey("password") && !form.ContainsKey("role"))
        return Results.Unauthorized();

    var login = form["login"];
    var role = form["role"];
    var password = form["password"];

    var person = people.FirstOrDefault(
        p => p.Login == login && p.Password == password && p.Role.Name == role);
    if (person is null) return Results.Unauthorized();

    var claims = new List<Claim>
    {
        new Claim(ClaimTypes.Name, person.Login),
        new Claim(ClaimTypes.Role, person.Role.Name)
    };
    ClaimsIdentity claimsIdentity = new ClaimsIdentity(claims,
        CookieAuthenticationDefaults.AuthenticationScheme);
    await context.SignInAsync(new ClaimsPrincipal(claimsIdentity));

    return Results.LocalRedirect(returnUrl ?? "/");
}
```

Рис. 3.13. Створення куки

Тепер перевіримо роботу даного застосунку. Відкриємо вебсторінку та спробуємо додати запис.

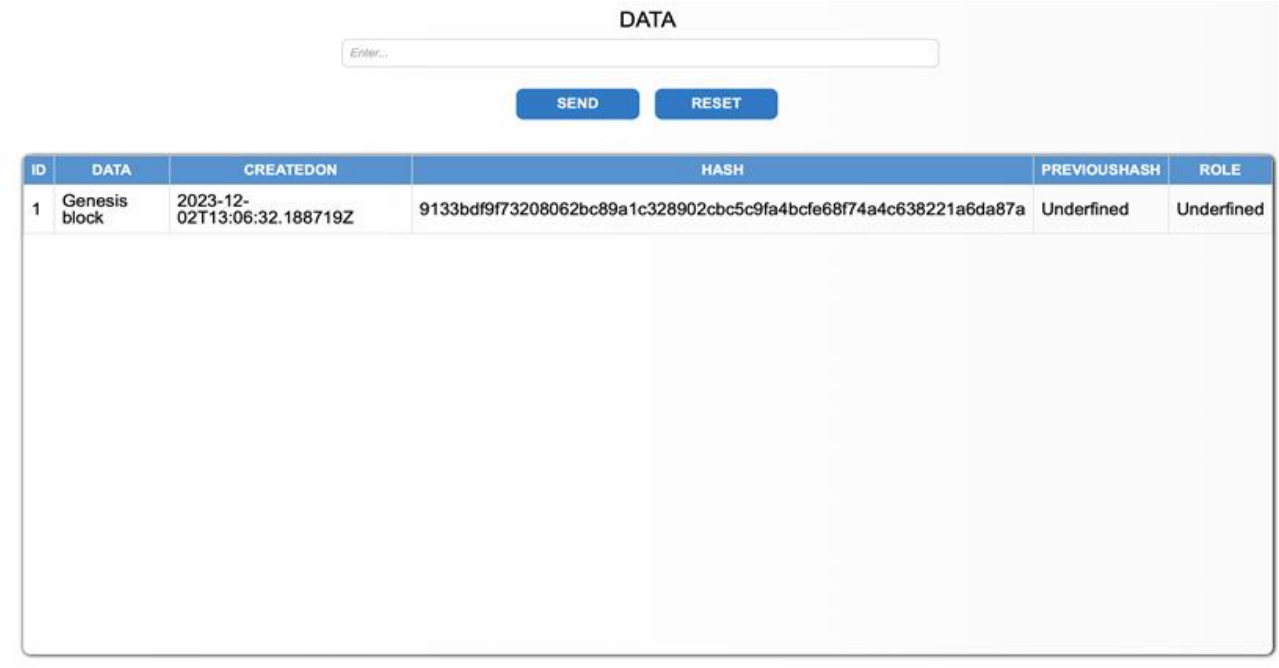


Рис. 3.14. Домашня вебсторінка

Так як ми не авторизовані, нас перенаправляє на сторінку авторизація та автентифікації.

Рис. 3.15. Форма автентифікації та авторизації

Введемо свої дані та автентифікуємось.

Рис. 3.16. Заповнена форма автентифікації та авторизації

Після вдалої автентифікації спробуємо додати запис.

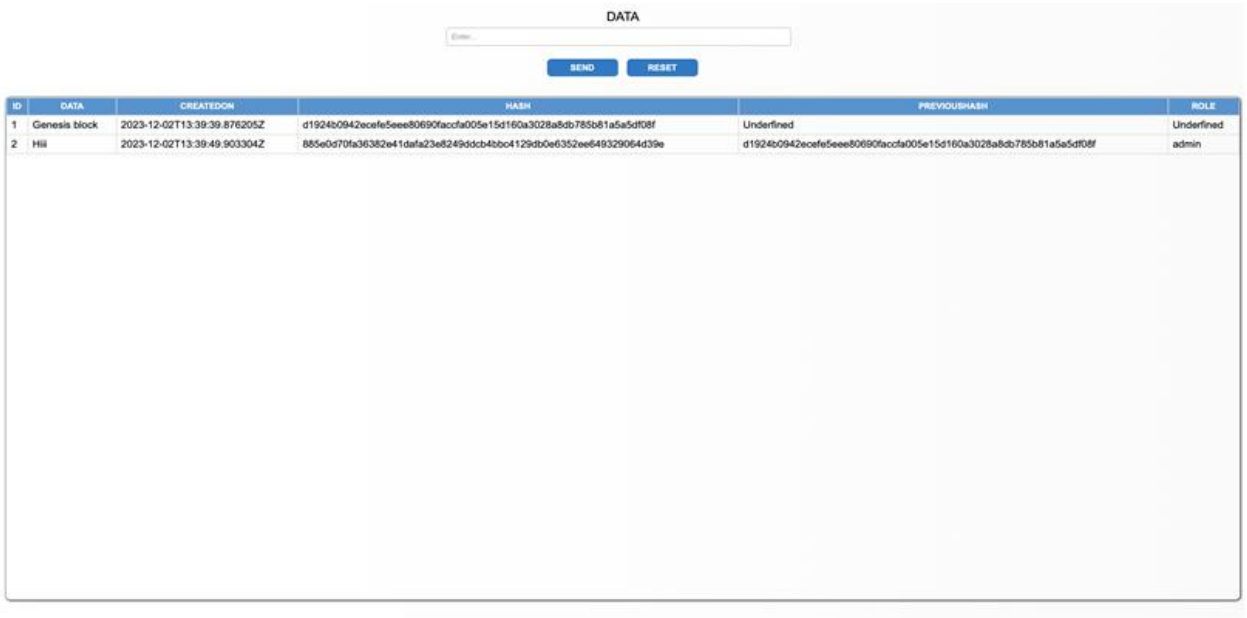


Рис. 3.17. Домашня сторінка

3.4. Висновки до розділу

На прикладі даного програмного забезпечення ми проаналізували роботу технології блокчейн та досягнення консенсусу. Виділили її основні властивості нписання програмного коду та вирішили поставлену нами раніше проблему збільшення розміру бази даних транзакцій. Дана технологія здійснює перетворення, видаляючи посередників і запроваджуючи протоколи консенсусу.

З даної роботи можна зрозуміти, що починає вимальовуватися новий світ. Після того як ми успішно увійшли у світ прозорості з децентралізованими, автономними та миттєво діючими ЗМІ, а потім у світ колективного користування з соціальними мережами та простором для спільної роботи, ми бачимо сьогодні загальну довіру, що виникає і міцніє, і нові варіанти. його застосування завдяки технології блокчейну.

Можливо, блокчейну вдасться зробити те, чого не вдалося досягти Інтернету: більшої гуманізації, свободи та довіри, що йдуть із самого серця цієї технології, і поступово ввести нас у світ нової економіки.

РОЗДІЛ 4. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

4.1. Глобальна кліматична проблема

За всю історію існування Землі клімат змінювався багато разів. Вченим відомо про 7 льодовикових періодів, після яких завжди наступало потепління[31].

Потепління в наш час – не лише природний процес, бо відбувається у 10 разів швидше, ніж будь-коли. Все частіше науковці вживають термін «кліматична криза» замість «зміни клімату», щоб підкреслити серйозність цієї проблеми та потребу її вирішувати вже зараз. Кліматична криза – це надмірно стрімка зміна клімату «через» підвищення глобальної середньої температури. Щоб протидіяти кліматичній кризі, слід досягти вуглецевої нейтральності вже 2050 року та адаптуватися до змін клімату[31].

Причини зміни клімату:

1. Парниковий ефект - це процес, за якого парникові гази затримують сонячну енергію на поверхні Землі та в атмосфері і перешкоджають її поверненню назад у космос. Парниковий ефект підтримує на Землі комфортну для життя температуру. Якби не було цього ефекту, то середня глобальна температура була б не +15°C, а -18°C[31].

2. Викиди парникових газів: Людство суттєво змінює концентрацію парникових газів в атмосфері, спалюючи викопне паливо: вугілля, нафту, газ тощо. Під час їх горіння вивільняється вуглець, який з’єднується з киснем у повітрі та утворює CO2[31].

За сотні тисяч років вперше в атмосфері сталося таке стрімке зростання вмісту CO2.

Наслідки зміни клімату в світі:

1. Глобальне потепління: Глобальна середня температура – це середнє значення всіх річних температур на Землі. Зазвичай дані обчислюються по регіонах за кожен день, а потім виводиться середнє арифметичне за рік для всієї планети. Різниця між річними показниками цих середніх температур і є те саме зростання (або падіння) середньої глобальної температури на Землі. Підвищення глобальної середньої температури на Землі означає, що спекотних днів у році стало більше, а холодних – менше. Це НЕ означає, що кожен день у порівнянні з відповідним днем року у доіндустріальну епоху став майже на 1 градус теплішим[32].

Згідно зі спостереженнями, середня глобальна температура на Землі вже зросла на 0,95°C з 1880 року. Глобальне потепління відбувається нерівномірно по планеті. Середня температура в арктичних регіонах планети вже зросла на 2°C[31, 32].

2. Танення льодовиків: Потепління в Арктиці відбувається вдвічі швидше у порівнянні з іншими регіонами планети. Тому льодовики тануть швидше. З 1979 року (перший повний рік супутникового спостереження) об’єм льоду в найтепліший сезон в Арктиці зменшився на 32%. За такої тенденції до середини століття в літній період Арктика буде без льоду[32].

3. Танення льодовиків має декілька серйозних наслідків:

- ✓ Скорочується площа білого покриву, який відбиває від 20% до 50% сонячної радіації. А площа океану збільшується та поглинає більше 95%. Так вода ще більше нагрівається і пришвидшує танення льодовиків, призводячи до більших змін клімату[31].
- ✓ За підрахунками вчених з National Snow and Ice Data Center, вічна мерзлота утримує 1 400 гігатонн вуглекислого газу – це майже вдвічі більше, ніж зараз містить атмосфера. Поки вічна мерзлота тане, вона поступово вивільняє ці поклади газу. Разом із CO_2 в атмосферу потрапляє Метан (CH_4) – газ із парниковим ефектом у 84 рази сильнішим ніж CO_2[31, 32].

- ✓ Підвищення рівня Світового океану. Вже зараз під водою зникають острови: Мальдіви, Фіджі, Сейшельські Острови, Маршаллові острови, Канарські острови, Федеративні Штати Мікронезії, Французька Полінезія, Філіппіни, Тувалу, Соломонові острови (вже втратили 5 островів через підняття рівня океану)[31, 32].

4. Хвилі тепла: Тренд, який фіксують науковці протягом останніх десятиліть, – хвилі тепла. Вони стають більш розповсюдженими у світі, тривають довше і стають більш екстремальними. Такою, наприклад, стала хвиля тепла влітку 2019 року у Європі[32].

5. Зміни в опадах: Підвищення температури збільшує випаровування та спричиняє перерозподіл вологи. Як наслідок, в одних регіонах випаровується надмірна кількість вологи та посилюється посуха. В інших регіонах ця волога конденсується, і там частішають зливи та шторми, що викликає ризики затоплення[32].

6. Зникнення біорізноманіття: Біорізноманіття – це розмаїття живих організмів на Землі; сюди входить різноманітність всередині видів, між видами та екосистемами. Через зміну клімату та людську діяльність за останні півстоліття чисельність популяцій хребетних тварин на Землі зменшилась на 68%. Це загрожує людству втратами рослинної і тваринної їжі, води, палива, ліків[31].

4.2. Адаптація до зміни клімату

Адаптація до зміни клімату – це пристосування природних чи людських систем до фактичних або очікуваних кліматичних впливів чи їхніх наслідків. Вона дозволяє знизити шкоду та скористатися можливостями, такими як створення нових робочих місць або економія коштів на ліквідацію наслідків надзвичайних ситуацій[32].

Адаптація до змін клімату може відбуватися на будь-якому рівні суспільства, від особистості до національного та міжнародного рівня.

Заходи з адаптації мають різні форми та формати та залежать від унікального контексту громади, країни чи регіону. Не існує універсального рішення – адаптація може варіюватися від побудови засобів захисту від повені, створення систем раннього попередження для циклонів і переходу на посухостійкі культури[31, 32].

Для кожної країни важливо створювати свої політики з адаптації. Їхня мета – зменшення вразливості до наслідків зміни клімату. Оскільки прояви зміни клімату є дуже різними, то і заходи, і політики з адаптації розробляються з урахуванням особливостей конкретної країни і галузі[31].

Можливими прикладами з адаптації до зміни клімату є: адаптація будівельних норм до майбутніх кліматичних умов та екстремальних погодних явищ; побудова та підвищення рівня дамб для захисту від повеней; розвиток посухостійких сільськогосподарських культур; створення систем раннього попередження циклонів[31].

Прикладом адаптації є місто Арнем, Нідерланди, поставило собі за мету протягом наступних 10 років зняти 10% асфальтового покриття та замінити його газонами, кущами та деревами. Це дозволить 90% дощової води вільно потрапити в ґрунт і запобігти затопленню доріг, тротуарів під час сильних злив. Також зелені зони сприяють зменшенню температури довкола[31].

Газони або дерева також можна висаджувати на дахах будинків. У Данії ще у 2010 році вирішено, що новобудови та модернізовані будинки з пласкими дахами повинні ставати зеленими. В одному лише Копенгагені вже нараховується понад 40 таких об’єктів[31].

Одним з найважливіших наслідків зміни клімату є підняття рівня моря. Острівні держави, такі як Кірібати, Фіджі, Маршалові острови вже зазнають значного впливу від збільшення рівня моря. Європейські країни – Нідерланди, Британія, Грецькі острови також знаходяться в зоні ризику[32].

Як один з прикладів адаптації влада Фіджі використовує поєднання мангрових лісів, які за рахунок потужної кореневої системи ефективно зменшують енергію хвиль та захищають ґрунт від ерозії. Інший спосіб – це будівництво морських стін, що також убезпечують місцевих жителів під час надзвичайних ситуацій[32].

4.3. Висновки до розділу

Сьогодні можна зі впевненістю сказати, що значні кліматичні зміни вже відбуваються. Ми повинні замислитися та зрозуміти, що людство не має права використовувати атмосферу планети для забруднення. Якщо ми не розпочнемо активно діяти, то вже незабаром наблизимось до тої межі, коли глобальну зміну клімату зупинити буде вже неможливо і життя на планеті у майбутньому буде під загрозою.

ВИСНОВКИ

В ході даної дипломної роботи було проаналізовано криптографічні методи шифрування, хеш-функції, модифікації технології блокчейн та баз даних, а також:

- зроблено порівняльний аналіз за класифікацією методів захисту даних, що дав змогу виявити її можливості для реалізації захисту даних в інформаційних системах за допомогою технології блокчейн;
- зроблено аналіз методів захисту баз даних та технології блокчейн, який показав необхідність застосування централізованої серверної бази даних;
- розроблено та протестувано програмний модуль захисту даних в ІС на основі технології блокчейн, який може використовуватись на комп'ютерах з обмеженим обсягом пам'яті та забезпечує збереження даних від несанкціонованого спотворення бази даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Савюк О. В. Програмний модуль забезпечення захисту даних у інформаційних системах на основі технології блокчейн : дипломна робота. Київ : НАУ, 2022. 85 с.

2. Про захист інформації в інформаційно-телекомунікаційних системах [Текст] : Закон України 1089-IX від 16.12.2020 / Верховна Рада // Відомості Верховної Ради України. – 1994. – 36. – 286.

3. Захист інформації. Технічний захист інформації. Загальні положення [Текст] : ДСТУ 3396.0-96 – 1996. – Чин. 1997.01.01. – К. : ДСТСЗІ СБ України, 1997, - 6с.

4. ISO/IEC 27001 : 2013 [Електроний ресурс] // ISO/IEC. - 2013. - Режим доступу до ресурсу: https://pqm-online.com/assets/files/pubs/translations/std/isomek-27 001-2013(rus).pdf.

5. Про захист персональних даних [Текст] : Закон України 2297-VI / Верховна Рада // Відомості Верховної Ради України. – 2010. – 34. – 481.

6. Л. Лелу Блокчейн от А до Я / Лоран Лелу : Эксмо, 2015. – 190 с.

7. П. Винья Эпоха криптовалют / Пол Винья, Майкл Кейси : Манн, Иванов и Фербер, 2018. – 432 с.

8. П. Винья Машина правды. Блокчейн и будущее человечества / Пол Винья, Майкл Кейси : Манн, Иванов и Фербер, 2018. – 320 с.

9. А. Табернакулов Блокчейн на практике / Александр Табернакулов, Ян Койфманн : Альпина Паблишер, 2019. – 264 с.

10. А. Цихилов Блокчейн. Принципы и основы / Александр Цихилов : Альпина PRO, 2019. – 192 с.

11. Н. Прасти Блокчейн. Разроботка приложений / Нараян Прасти : БХВ-Петербург, 2018. – 256 с.

12. . Компанія Prypto Биткойн для чайников / Компанія Prypto : Диалектика, 2017. – 240 с.

13. И. Башир Блокчейн. Архитектура, криптовалюты, инструменты разработки, смарт-контракты / Имран Башир : Print2print, 2019. – 538 с.

14. А. Цихилов Блокчейн. Принципы и основы / Александр Цихилов : Альпина Паблишер, 2019. – 188 с.

15. С. Равал Децентрализованые приложения. Технология Blockcahin в действии / Сирадж Равал : Питер, 2017. – 192 с.

16. Н. Поппер Цифровое золото: невероятная история Биткойна, или как идеалисты и бизнесмены изобретают деньги заново. / Натаниел Поппер : Науковий світ, 2023. – 370 с.

17. К. Скиннер ValueWeb. Как финтех-компании используют блокчейн и мобильные технологии для создания интернета цен / Крис Скиннер : МИФ, 2017. – 416 с.

18. П. Винья Эпоха криптовалют. Как биткойн и блокчейн меняют мировой экономический порядок / Пол Винья, Майкл Кейси : МИФ, 2018. – 432 с.

19. И. Башир Блокчейн. Архитектура, криптовалюты, инструменты разработки, смарт-контракты / Имар Башир : Print2print, 2019. – 538 с.

20. А. Рябых Как заработать на криптовалютах и блокчейне. Объясняем на пальцах / Андрей Рябых, Светлана Русова : Print2print, 2019. – 256 с.

21. М. Свон Блокчейн. Схема ново економики / Мелани Свон : Киндл, 2015. – 152 с.

22. Н. Прасти Блокчейн. Разроботка приложений / Нараян Прасти : БХВ-Петербург, 2018. – 256 с.

23. Blockchain C# | Реализация Блокчейн C# [Електроний ресурс]. – Режим доступу: https://shwanoff.ru/blockchain/

24. Blockchain Explained [Електроний ресурс]. – Режим доступу: https://www.investopedia.com/terms/b/blockchain.asp

25. Whait is blockchain technology? [Електроний ресурс]. – Режим доступу: https://www.ibm.com/topics/what-is-blockchain

26. What is blockchain technology? How does it work? [Електроний ресурс]. – Режим доступу: https://builtin.com/blockchain

27. Blockchain – The new technology of trust? [Електроний ресурс]. – Режим доступу: https://www.goldmansachs.com/insights/pages/blockchain/

28. History of blockchain [Електроний ресурс]. – Режим доступу: https://www.icaew.com/technical/technology/blockchain-and-cryptoassets/blockchain-articles/what-is-blockchain/history

29. Что такое технология блокчейн? [Електроний ресурс]. – Режим доступу:https://aws.amazon.com/ru/what-is/blockchain/?aws-products-all.sort-by=item.additionalFields.productNameLowercase&aws-products-all.sort-order=asc

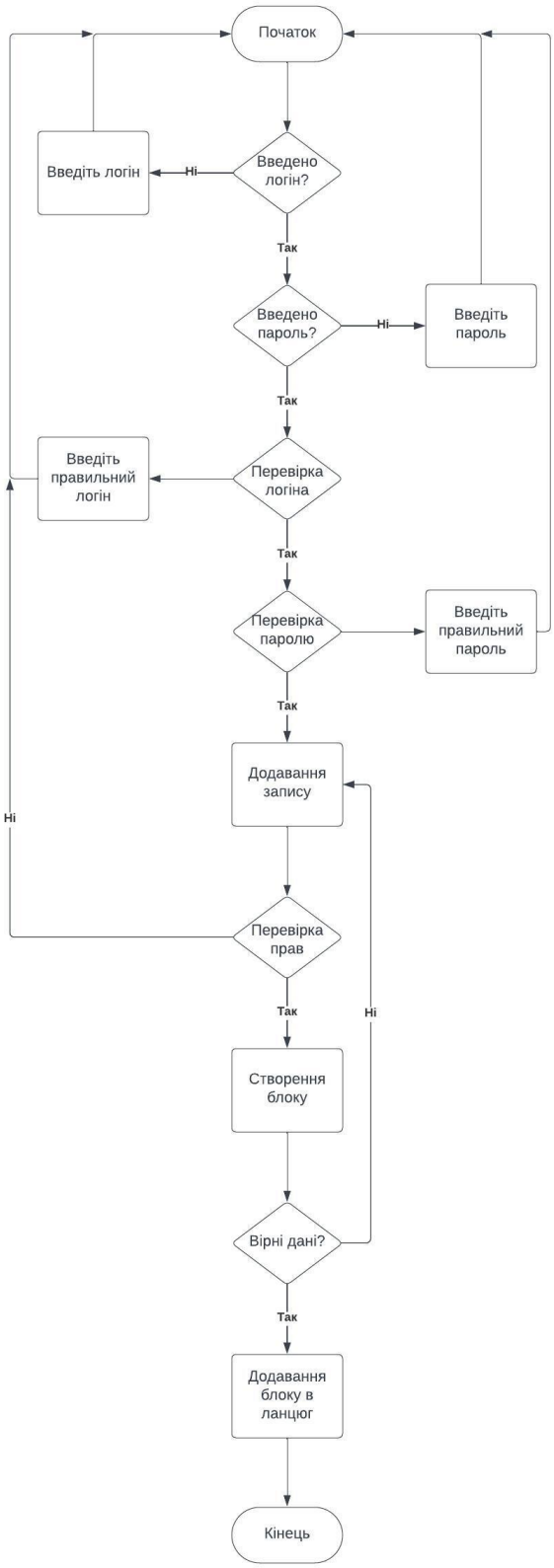
30. Что такое блокчейн? [Електроний ресурс]. – Режим доступу: https://www.oracle.com/cis/blockchain/what-is-blockchain/

31. Фейген Б. Велике потепління: зміна клімату та піднесення й гибель цивілізацій / Б. Фейген; пер. з англ. Т. Цимбала. – Київ: Ніка-Центр, 2016. – 272 с.

32. Бабич А. О. Засуха, суховій і пилова буря в період глобальних змін клімату. Т. 1 / А. О. Бабич, А. А. Бабич-Побережна. – Вінниця: Діло, 2014. – 468 с.

Додаток А

Алгоритм розробленого програмного модуля



Додаток Б

Фрагмент вихідного коду програмного модуля

Серверна частина

Клас ролі об’єкта

```
namespace WebBlockchain;

public class Role
{
    public Role(string name)
    {
        Name = name;

        Validation(this);
    }

    // Назва ролі
    [Required]
    [StringLength(50, MinimumLength = 1)]
    public string Name { get; private set; }

    // Валідація об’єкта
    private void Validation(Role role)
    {
        var results = new List<ValidationResult>();
        var context = new ValidationContext(role);

        if (!Validator.TryValidateObject(role, context, results, true))
        {
```

Продовження додатку Б

```
            foreach (var error in results)
                throw new Exception(error.ErrorMessage);
        }
    }
}
```

Клас об’єкта

```
namespace WebBlockchain;

public class Person
{
    public Person(string login, string password, Role role)
    {
        Login = login;
        Password = password;
        Role = role;

        Validation(this);
    }

    // Логін користувача
    [Required]
    [StringLength(50, MinimumLength = 1)]
          public string Login { get; private set; }
    // Пароль користувача
    [Required]
    [StringLength(50, MinimumLength = 1)]
    public string Password { get; private set; }
    // Роль користувача
```

Продовження додатку Б

```
    [Required]
        public Role Role { get; private set; }
    // Валідація об’єкту
        private void Validation(Person person)
        {
        var results = new List<ValidationResult>();
        var context = new ValidationContext(person);

        if (!Validator.TryValidateObject(person, context, results, true))
        {
            foreach (var error in results)
                throw new Exception(error.ErrorMessage);
        }
    }
}
```

Клас блоку ланцюга

```
namespace WebBlockchain;

public class Block
{
    public Block()
    {
        Id = 1;
        Data = "Genesis block";
        CreatedOn = DateTime.UtcNow;
        PreviousHash = "Underfined";
        Role = "Underfined";
        Hash = GetHashUseSHA256(GetConcatedString());
```

Продовження додатку Б

```
}

public Block(string data, string role, Block previousBlock)
{
    Id = previousBlock.Id + 1;
    Data = data;
    CreatedOn = DateTime.UtcNow;
    PreviousHash = previousBlock.Hash;
    Role = role;
    Hash = GetHashUseSHA256(GetConcatedString());

    Validation(this);
}

// Номер запису
[Required]
public int Id { get; private set; }
// Інформація запису
[Required]
[StringLength(50, MinimumLength = 1)]
public string Data { get; private set; }
// Дата і час створення запису
[Required]
public DateTime CreatedOn { get; private set; }
// Хєш запису
[Required]
public string Hash { get; private set; }
// Хєш попереднього запису
```

Продовження додатку Б

```
[Required]
public string PreviousHash { get; private set; }
// Роль автора запису
[Required]
public string Role { get; private set; }

// Валідація об’єкта
private void Validation(Block block)
{
    var results = new List<ValidationResult>();
    var context = new ValidationContext(block);

    if (!Validator.TryValidateObject(block, context, results, true))
    {
        foreach (var error in results)
            throw new Exception(error.ErrorMessage);
    }
}

// Функція для конкатанації строк
private string GetConcatedString()
{
    StringBuilder concatedString = new StringBuilder();

    concatedString.Append(Id.ToString());
    concatedString.Append(Data);
    concatedString.Append(CreatedOn.ToString()); //Change format!
    concatedString.Append(PreviousHash);
```

```
        concatedString.Append(Role);

        return concatedString.ToString();
    }

    // Функція для створення хєшу об’єкта
    private string GetHashUseSHA256(string concatedString)
    {
        using SHA256 sha256 = SHA256.Create();
        StringBuilder hash = new StringBuilder();

        var message = Encoding.ASCII.GetBytes(concatedString);
        var hashValue = sha256.ComputeHash(message);

        foreach (byte x in hashValue)
            hash.Append(string.Format("{0:x2}", x));

        return hash.ToString();
    }
}
```

Клас ланцюга

```
namespace WebBlockchain;

public class Chain
{
    public Chain() => CreateGenesisBlock();

    public Chain(Block block)
```

Продовження додатку Б

```
{
    if (Blocks.Count == 0) CreateGenesisBlock();

    Blocks.Add(block);
}

// Список блоків
private List<Block> Blocks { get; set; } = new List<Block>();
// Попередній блок
private Block PreviousBlock { get; set; }

// Метод для створення генезіс блоку
private void CreateGenesisBlock()
{
    var genesisBlock = new Block();

    Blocks.Add(genesisBlock);
    PreviousBlock = genesisBlock;
}

// Метод для створення та додавання запису в список
public void Add(string data, string role)
{
    var block = new Block(data, role, PreviousBlock);

    Blocks.Add(block);
    PreviousBlock = block;
}
```

Продовження додатку Б

```
    // Метод для перевірки коректності блокчейну
    public bool CheckChain()
    {
        var previousHash = new Block().Hash;

        foreach (var block in Blocks.Skip(1))
        {
            if (previousHash != block.PreviousHash)
                return false;

            previousHash = block.PreviousHash;
        }

        return true;
    }

    // Метод, що повертає список
    public List<Block> GetAllRecords() => Blocks;
}
```

Основний клас серверної частини

```
var people = new List<Person>
{
    new Person("alex@gmail.com", "12345", new Role("admin")),
    new Person("tom@gmail.com", "678910", new Role("user")),
    new Person("guest@gmail.com", "12345", new Role("guest"))
};
```

Продовження додатку Б

```
var builder = WebApplication.CreateBuilder();

builder.Services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationSc
heme)
    .AddCookie(options =>
    {
        options.LoginPath = "/signin";
        options.AccessDeniedPath = "/accessdenied";
    });
builder.Services.AddAuthorization();
builder.Services.AddSingleton<Chain>();

var app = builder.Build();

app.UseDefaultFiles();
app.UseStaticFiles();
app.UseMiddleware<ErrorHandlingMiddleware>();

app.UseAuthentication();
app.UseAuthorization();

app.MapGet("/accessdenied", async (context) =>
{
    context.Response.ContentType = "text/html; charset=utf-8";
    await context.Response.WriteAsync("Access denied!");
});

app.MapGet("/signin", async (context)
```

Продовження додатку Б

```
   => await SendHtmlFileHandler("wwwroot/sign_in.html", context));
app.MapPost("/signin", PostSignInHandler);

app.MapGet("/signup", async (context)
   => await SendHtmlFileHandler("wwwroot/sign_up.html", context));
app.MapPost("/signup", PostSignUpHandler);

app.MapGet("/logout", GetLogOutHandler);

app.MapGet("/api/data", (Chain chain) => chain.GetAllRecords());
app.MapPost("/api/data", PostApiDataHandler);

app.MapGet("/", async (context)
   => await SendHtmlFileHandler("wwwroot/index.html", context));

app.Run();

async Task<IResult> PostSignUpHandler(string? returnUrl, HttpContext context)
{
   var form = context.Request.Form;

   if (!form.ContainsKey("login") && !form.ContainsKey("password")
&& !form.ContainsKey("role"))
       return Results.StatusCode(402);

   var login = form["login"];
   var password = form["password"];
   var role = form["role"];
```

Продовження додатку Б

```
    people.Add(new Person(login, password, new Role(role)));

    return Results.LocalRedirect(returnUrl ?? "/");
}

async Task<IResult> GetLogOutHandler(HttpContext context)
{
    await
context.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme);
    return Results.LocalRedirect("/signin");
}

async Task<IResult> PostSignInHandler(string? returnUrl, HttpContext context)
{
    var form = context.Request.Form;

    if (!form.ContainsKey("login") && !form.ContainsKey("password")
&& !form.ContainsKey("role"))
        return Results.Unauthorized();
    var login = form["login"];
    var role = form["role"];
    var password = form["password"];

    var person = people.FirstOrDefault(
        p => p.Login == login && p.Password == password && p.Role.Name == role);
    if (person is null) return Results.Unauthorized();
```

Продовження додатку Б

```
    var claims = new List<Claim>
    {
        new Claim(ClaimTypes.Name, person.Login),
        new Claim(ClaimTypes.Role, person.Role.Name)
    };
    ClaimsIdentity claimsIdentity = new ClaimsIdentity(claims,
        CookieAuthenticationDefaults.AuthenticationScheme);
    await context.SignInAsync(new ClaimsPrincipal(claimsIdentity));

    return Results.LocalRedirect(returnUrl ?? "/");
}
//[Authorize(Roles = "admin, user")]
async Task<IResult> PostApiDataHandler(HttpContext context, Chain chain)
{
    var form = context.Request.Form;
    chain.Add(form["data"], form["role"]);

    return Results.LocalRedirect("/");
}

async Task SendHtmlFileHandler(string fileName, HttpContext context)
{
    context.Response.ContentType = "text/html; charset=utf-8";
    await context.Response.SendFileAsync(fileName);
}
```

Клієнтська частина

Основна HTML сторінка

```
<!DOCTYPE html>
```

Продовження додатку Б

```
<html lang="en">

<head>
  <title>Document</title>
  <link rel="stylesheet" href="css/style.css">
  <script src="js/scrypt.js"></script>
  <meta http-equiv="Content-type" content="text/html;charset=UTF-8" />
</head>

<body>
  <div class="enter_data">
    <form action="/api/data" method="post">
      <div class="inputData">
        <p>Data</p>
        <input tabindex="1" type="text" name="data" placeholder="Enter..." />
      </div>
      <div class="buttons">
        <button tabindex="4" type="submit">Send</button>
        <button tabindex="5" type="reset">Reset</button>
      </div>
    </form>
  </div>
  <div id="table-container" class="table_data"></div>
</body>

</html>
```

HTML сторінка авторизації та аутентифікації

```
<!DOCTYPE html>
```

Продовження додатку Б

```
<html lang="en">

<head>
  <title>Document</title>
  <link rel="stylesheet" href="css/sign_in_style.css">
  <meta http-equiv="Content-type" content="text/html;charset=UTF-8" />
</head>

<body>
  <div class="main_block">
    <form action="/singin" method="post">
      <div>
        <p>Login</p>
        <input tabindex="1" type="text" name="login" placeholder="Enter...">
      </div>
      <div class="radio">
        <p>Role</p>
        <div>
          <span class="figure">&#x1F510</span>
          <input tabindex="2" type="radio" name="role" value="admin" id="admin"
/>
          <label for="admin">Admin</label>
        </div>
        <div>
          <span class=«figure">&#x1F64E</span>
          <input checked tabindex="3" type="radio" name="role" value="user"
id="user" />
          <label for="user">User</label>
```

Продовження додатку Б

```
          </div>
          <div>
            <span class="figure">&#x1F64E</span>
            <input checked tabindex="3" type="radio" name="role" value="guest"
id="guest" />
            <label for="guest">Guest</label>
          </div>
        </div>
        <div>
          <p>Password</p>
          <input tabindex="2" type="password" name="password"
placeholder="Enter...">
        </div>
        <div class="buttons">
          <button tabindex="4" type="submit">Send</button>
          <button tabindex="5" type="reset">Reset</button>
        </div>
      </form>
    </div>
</body>

</html>
```

JS-скрипт взаємодії обох частин вебдодатку

```
document.addEventListener('DOMContentLoaded', async function () {
  try {
    // Отримання JSON даних з сервера
    const response = await fetch('https://localhost:7209/api/data');
    const jsonData = await response.json();
```

Продовження додатку Б

```
// Отримання контейнера для таблиці
const tableContainer = document.getElementById('table-container');

// Створення таблиці
const table = document.createElement('table');
table.classList.add("main-table");

// Створення заголовка таблиці
const header = table.createTHead();
const headerRow = header.insertRow(0);
headerRow.classList.add("header-2");
for (const key in jsonData[0]) {
  const th = document.createElement('th');
  th.innerHTML = key;
  headerRow.appendChild(th);
}

// Додавання даних до таблиці
for (let i = 0; i < jsonData.length; i++) {
  const row = table.insertRow(i + 1);
  for (const key in jsonData[i]) {
    const cell = row.insertCell();
    cell.innerHTML = jsonData[i][key];
  }
}

// Вставка таблиці у контейнер
```

Продовження додатку Б

```
    tableContainer.appendChild(table);
  } catch (error) {
    console.error('Помилка при отриманні або обробці даних:', error);
  }
});
```