ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Пояснювальна записка

до магістерської роботи
на ступінь вищої освіти магістр

на тему: **«Методи шифрування текстової інформації за допомогою блокчейн технології»**

Виконав: студент 6 курсу, групи __ПДМ-61
Спеціальності 121_Інженерія програмного забезпечення
(шифр і назва спеціальності)

_____Дзима А.В._____
(прізвище та ініціали)

Керівник_____Щербина І.С._____
(прізвище та ініціали)

Рецензент_____
(прізвище та ініціали)

Нормоконтроль _____

Київ – 2020

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення
Ступінь вищої освіти - «Магістр»
Спеціальність – 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри
інженерії програмного забезпечення
_____О.В. Негоденко
"____" _____2020 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Дзима Андрій Вікторович
(прізвище, ім'я, по батькові)

1.Тема роботи: «Методи шифрування текстової інформації за допомогою блокчейн технології»
Керівник роботи Щербина Ірина Сергіївна к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від 13.10. 2020 року № 230 .
2. Строк подання студентом роботи 24.12.2020
3. Вихідні дані до роботи: Алгоритми шифрування, математичні основи блокчейн, архітектура блокчейн, фреймворк "Django".

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Блокчейн технології для шифрування текстової інформації
2. Математична складова шифрування текстової інформації за допомогою блокчейн технологій
.
3. Розробка архітектури блокчейн для шифрування текстової інформації
.
4. Застосування методів шифрування текстової інформації за допомогою блокчейн технологій

5. Перелік графічного матеріалу

1. Мета та завдання магістерської роботи;
2. Актуальність магістерської роботи;
3. Роль блокчейн технологій в шифруванні текстової інформації;
4. Алгоритми блокчейн технологій;
5. Математична складова блокчейн методів шифрування;
6. Поняття архітектури блокчейн;
7. Розробка архітектури блокчейн;
8. Складові методів блокчейн технологій для шифрування;
9. Реалізація алгоритмів блокчейн шифрування;
10. Створення панелі обміну для інформації на базі фреймворку Django;
11. Висновки та апробація результатів магістерської роботи.

6. Дата видачі завдання__02.11.2020_____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	02.11-06.11.2020	
2	Блокчейн технології для шифрування текстової інформації	06.11-09.11.2020	
3	Математична складова шифрування текстової інформації за допомогою блокчейн технологій	09.11-12.11.2020	
4	Розробка архітектури блокчейн для шифрування текстової інформації	12.11-16.11.2020	
5	Застосування методів шифрування текстової інформації за допомогою блокчейн технологій	16.11-27.11.2020	
6	Розробка обов'язкових демостраційних матеріалів	09.12.2020	
7	Попередній захист роботи	14.12-18.12.2020	
8	Здача Здача роботи в деканат	24.12.2020	

Студент _____ _____
(підпис) (прізвище та ініціали)

Керівник роботи _____ _____
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської роботи 66 с., 35 рис., 15 джерел.

Об’єкт дослідження – технології блокчейн для захисту текстової інформації.

Предмет дослідження – Методи блокчейн технологій для шифрування текстової інформації.

Мета роботи – Захист текстової інформації за допомогою блокчейн технологій.

Методи дослідження – алгоритми шифрування, блокчейн, фреймворк Django.

Актуальність роботи – використання актуальних на даний час методів шифрування текстової інформації за допомогою блокчейн технологій, з детальним описом процесу шифрування та результатом виконаних дій.

Стислий опис результатів: шифрування необхідного тексту, завдяки використанням сучасних методів захисту текстової інформації за допомогою блокчейн технологій, перевірка на працездатність та захищеність.

Ключові слова: криптографія, месенджер, кодування, хеш, блокчейн, смарт-контракти, віртуальна машина, RSA, валідація, верифікація.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Криптографія - наука про методи забезпечення конфіденційності, цілісності даних, аутентифікації, шифрування;

Месенджер - програма з обміну миттєвими повідомленнями;

Кодування - операція заміни коду текстових даних; заміна звичайних текстових даних скороченими умовними позначеннями; переклад будь-якої інформації, вираженої засобами природної мови, в послідовність умовних символів, сигналів за певними правилами, що називають кодом;

Хеш - функція, або функція згортки - функція, що здійснює перетворення масиву вхідних даних довільної довжини в бітову рядок встановленої довжини, що виконується певним алгоритмом;

Блокчейн - побудований за певними правилами безперервний послідовний ланцюжок блоків (зв'язний список), що містить інформацію;

Смарт-контракти - комп'ютерний алгоритм, призначений для формування, контролю і надання інформації про володіння чим-небудь;

Ethereum - платформа для створення децентралізованих онлайн-сервісів на базі блокчейна (децентралізованих додатків), що працюють на базі розумних контрактів;

Віртуальна машина - програмна і / або апаратна система, що емулює апаратне забезпечення деякої платформи, що буде взаємодіяти з апаратним комплексом;

RSA - криптографічний алгоритм з відкритим ключем, який базується на обчислювальній складності задачі факторизації великих цілих чисел.

Валідація - доказ того, що вимоги конкретного користувача, продукту, послуги або системи задоволені;

Верифікація - підтвердження на основі наданням об'єктивних доказів того, що встановлені вимоги були виконані.

ВСТУП

В час стрімкого розвитку мережі інтернет все частіше постає питання безпечної передачі текстової інформації між користувачами. Пересилання текстової інформації через мережу інтернет - поширена ситуація, а захист таких даних відіграє дуже важливу роль в функціонуванні великої кількості компаній. На даний час, існує ряд варіантів передачі текстової інформації, які потребують належного рівня захисту в процесі передачі. Методи передачі та шифрування залежать від загальних потреб відправника та отримувача.

По-перше, надійний захист текстової інформації важливий для розвитку в різних напрямках новітніх технологій.

По-друге, забезпечити безпечну передачу особистих даних між звичайними користувачами в мережі.

По-третє, зробити безпечною роботу фінансових операцій та систем. За даними Identity Theft Resource Center (ITRC), в 2019 році було зафіксовано +1579 витоків даних, від яких постраждало приблизно 179 мільйонів записів. Виходить, що за один календарний рік число порушень даних зросла на 44%. Саме тому, розвиток методів шифрування текстової інформації за допомогою блокчейн технологій є надзвичайно актуальним.

В магістерській роботі буде описано методи шифрування текстової інформації за допомогою блокчейн технологій. Особлива увага буде приділена математичній та архітектурній базі блокчейн технологій для безпечної передачі інформації в мережі. Дослідження буде основане на власному досвіді, та задокументоване для подальшого ознайомлення з ним.

1 БЛОКЧЕЙН ТЕХНОЛОГІЇ ДЛЯ ШИФРУВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ

1.1 Шифрування даних як основа захисту текстової інформації

Шифрування - це спосіб приховування початкового сенсу повідомлення або іншого документу, що забезпечує спотворення його первинного вмісту. Перетворення звичайного, зрозумілого вмісту в код називається кодуванням.

При цьому мається на увазі, що є взаємна однозначна відповідність між символами тексту та коду - в цьому і полягає основна відмінність кодування від шифрування.

Часто кодування і шифрування помилково приймають за одно і теж, забувши про те, що для відновлення закодованого повідомлення, досить знати правило заміни, тоді як для розшифровки вже зашифрованого повідомлення крім знання правил шифрування, потрібно ключ до шифру. Під ключем в даному випадку мається на увазі конкретний секретний стан параметрів алгоритмів шифрування і дешифрування.

Зашифрувати можна не лише текст, але і різні дані - від файлів баз цих і текстових процесорів до файлів зображень.

1.2 Симетричне шифрування в блокчейн

Симетричне шифрування - це метод криптографії, в якому один ключ відповідає за шифрування і дешифрування даних. Сторони, що беруть участь, поділяють цей ключ, пароль або кодову фразу, і вони можуть використовувати його для дешифрування або шифрування будь-яких повідомлень.

Згідно з проектом захисту відкритих веб-додатків, деякі з найбільш поширених алгоритмів, що використовуються для симетричної криптографії, включають в себе стандарт шифрування даних (DES), який використовує 64-бітові ключі.

Потрійний DES, який тричі застосовує алгоритм DES з різними ключами і стандарт розширеного шифрування (AES), алгоритм, який Національний інститут стандартів і технологій США рекомендує використовувати для безпечного зберігання та передачі даних.

Симетричні ключові шифри або алгоритми, що використовуються для шифрування та дешифрування, є привабливими для організацій, оскільки вони недорогі, незважаючи на рівень захисту, який вони надають. Дійсно, аутентифікація вбудована в симетричну криптографію, оскільки сторони не можуть розшифровувати дані, зашифровані одним симетричним ключем, використовуючи інший симетричний ключ.

Центр знань IBM зазначає, що симетричні ключові шифри також менші за розміром. Ця властивість допомагає мінімізувати затримку часу, пов'язану з шифруванням і дешифруванням даних. Але симетричне шифрування не є досконалим.

Ключі в цьому методі криптографії живуть вічно, що означає, що організації повинні інвестувати в ведення журналу і аудит ключів протягом їх життєвого циклу.

Це також означає, що, якщо симетричний ключ втрачено, організації не зможуть його згадати. Замість цього вони повинні шифрувати та розшифровувати дані за допомогою іншого ключа, як тільки вони відновлять свої дані в незашифрованому вигляді.

Враховуючи бізнес-витрати, пов'язані з втратою симетричного ключа, компаніям необхідно проявляти велику обережність, щоб зацікавлені сторони надійно обміняли свій ключ.

На основі бази знань Венафі одна відповідь - це система тримання під вартою, за допомогою якої зберігачі отримують частини ключа з модуля апаратної безпеки (HSM) або фізичного обчислювального пристрою, який управляє ключами. Потім вони захищають ці компоненти і відправляють їх одержувачам, які вводять відповідні ключові фрагменти в HSM для формування ключа.

Успішне введення всіх компонентів дозволяє сторонам, що беруть участь, шифрувати і розшифровувати дані за допомогою завершеного ключа:

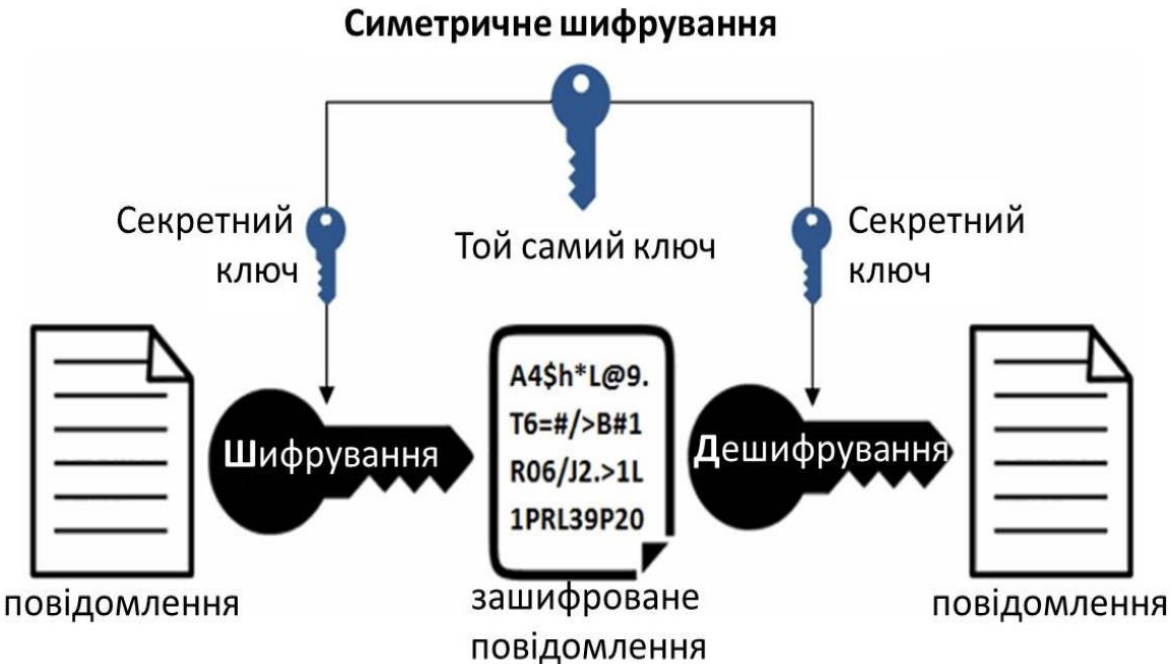


Рис. 1.1. - Схема симетричного шифрування

Крім того, зберігач може отримати симетричний ключ, загорнутий асиметричним сховищем ключів. Потім зберігач надсилає це сховище ключів одержувачу, який завантажує сховище ключів у HSM. Модуль, у свою чергу, розгортає сховище ключів, тим самим дозволяючи одержувачу шифрувати і розшифровувати повідомлення.

Звичайно, цей метод має свої межі. Якщо одержувачу завжди потрібен інший ключ для шифрування симетричного ключа, все може вийти з-під контролю і привести до нескінченного циклу ключів в залежності від додаткових ключів.

Коли все сказано і зроблено, організаціям потрібен спосіб контролювати свої ключі. Цей процес може стати ресурсомістким, оскільки, якщо кільком сторонам необхідно встановити свої власні захищені канали зв'язку один з одним з використанням симетричного шифрування, їм знадобляться власні ключі для кожного каналу. Саме тому в інтересах організацій автоматизувати управління ключами.

1.3 DES алгоритм шифрування як базова складова блокчейн технологій

DES - симетричний алгоритм шифрування, в якому один ключ використовується як для шифрування, так і для розшифрування даних.

DES розроблений фірмою IBM і затверджений урядом США в 1977 році як офіційний стандарт.

Алгоритм DES широко використовувався при зберіганні і передачі даних між різними обчислювальними системами, в поштових системах, в електронних системах креслень і при електронному обміні комерційною інформацією.

Стандарт DES реалізовувався як програмно, так і апаратно. Підприємствами різних країн був налагоджений масовий випуск цифрових пристроїв, що використовують DES для шифрування даних. Всі пристрої проходили обов'язкову сертифікацію на відповідність стандарту.

Алгоритм шифрування DES використовує ключ довжиною 64 біт. У той час вважалося, що випробувати всі 62 213 512 741 842 721 можливих ключів (сім з 16 нулями) було б неможливо, тому що комп'ютери не настільки потужнимі.

У 1998 році Electronic Frontier Foundation (EFF) створив спеціальну машину, яка могла розшифрувати повідомлення, випробувавши всі можливі ключі менш ніж за три дні. Машина коштувала приблизно 300000 Доларів США і шукала більше 77 мільярдів ключів в секунду.

Розмір блоку в DES-64 біта, для шифрування використовує ключ з довжиною 64 біт, кількість раундів – 16. DES є класичною мережею Фейштеля з двома гілками. За кілька раундів алгоритм перетворює 64-бітний вхідний блок даних в 64-бітний вихідний блок.

Стандарт DES побудований на комбінованому використанні перестановки, заміни і гамування. Дані для шифрування повинні бути представлені в двійковому вигляді.

Процес шифрування кожного 64-бітового блоку вихідних даних можна розділити на три етапи:

1. початкова підготовка блоку даних;

2. 16 раундів " основного циклу";

3. кінцева обробка блоку даних.

На першому етапі виконується початкова перестановка 64-бітного вихідного блоку даних. При початковій перестановці біти блоку даних певним чином переупорядковуються, що надає деяку "хаотичність" вихідного повідомлення, знижуючи можливість використання криптоаналізу статистичними методами.

Одночасно з початковою перестановкою блоку даних виконується початкова перестановка 64 біт ключа. У кожному з циклів використовується відповідний 48-бітний частковий ключ.

Ключі виходять за певним алгоритмом, використовуючи кожен з бітів початкового ключа по кілька разів. У кожному раунді 64-бітний ключ ділиться на дві 28-бітові частини.

Потім частини зсуваються вліво на один або два біт залежно від номера раунду. Після зсуву певним чином вибирається 48 з 64 бітів. Через те, що при цьому не тільки вибирається підмножина бітів, але і змінюється їх порядок, ця операція називається "перестановка зі стисненням". Її результатом є набір з 48 бітів.

В середньому кожен біт вихідного 64-бітного ключа використовується в 14 з 16 підключів, хоча не всі біти використовуються рівну кількість разів.

На другому етапі блок ділиться на дві гілки по 32 біта кожна і виконується основний цикл перетворення, організований по мережі Фейштеля і складається з 16 однакових раундів.

При цьому в кожному раунді виходить проміжне 64-бітне значення, яке потім обробляється в наступному раунді.

Спочатку права частина блоку збільшується до 48 бітів, використовуючи таблицю, яка визначає перестановку плюс розширення на 16 бітів. Ця операція приводить розмір правої частини у відповідність з розміром ключа для виконання операції XOR. За рахунок виконання цієї операції швидше зростає залежність всіх бітів результату від бітів вихідних даних і ключа. Саме тому потріно створити схему шифрування за алгоритмом DES, задля зручного орієнтування в процесі

шифрування необхідних даних шифрування текстової інформації задопомогою блокчейн технологій:

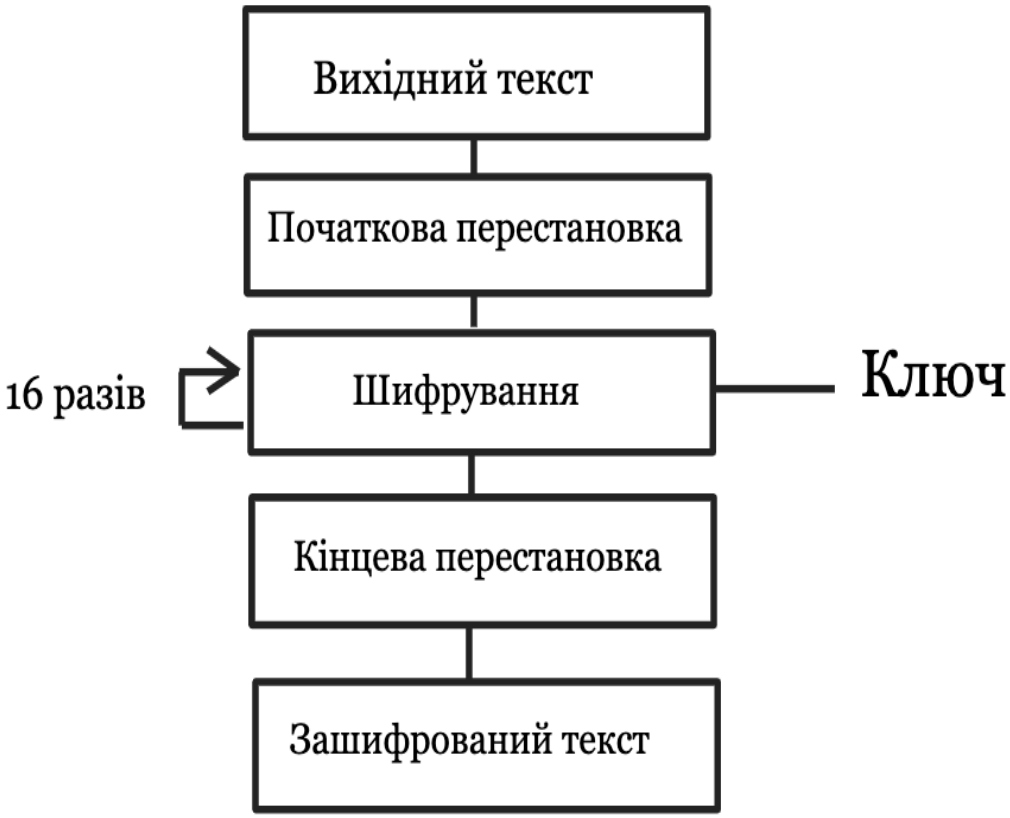


Рис. 1.2. - Схема симетричного шифрування за алгоритмом DES

Після виконання перестановки з розширенням для отриманого 48-бітного значення виконується операція XOR з 48-бітним підключенням. Потім отримане 48-бітове значення передається на вхід блоку підстановки, результат якої - 32-бітове значення.

Підстановка виконується у восьми блоках підстановки. При виконанні цієї операції 48 бітів даних поділяються на вісім 6-бітових подблоков, кожен з яких по своїй таблиці замінюється чотирма бітами.

Підстановка за допомогою S-блоків є одним з найважливіших етапів DES. Таблиці замін для цієї операції спеціально спроектовані так, щоб забезпечувати максимально можливу безпеку. В результаті цього етапу виходять вісім 4-бітових блоків, які знову об'єднуються в єдине 32-бітове значення.

Далі отримане 32-бітове значення обробляється за допомогою незалежної від використовуваного ключа перестановки. Метою перестановки є максимальне зміна бітів, щоб в наступному циклі шифрування кожен біт з великою ймовірністю оброблявся іншим блоком перестановки.

Результат перестановки об'єднується за допомогою операції XOR з лівою половиною початкового 64-бітового блоку даних. Далі ліва і права частини міняються місцями, і починається наступний раунд.

На другому етапі блок ділиться на дві частини (гілки) по 32 біта кожна. Права гілка перетворюється, використовуючи деяку функцію і відповідний частковий ключ, який виходить з основного ключа шифрування за спеціальним алгоритмом перетворення ключів. Далі проводиться обмін даними між лівою і правою гілками блоку. Це повторюється в циклі 16 разів.

Нарешті, на останньому третьому етапі проводиться перестановка результату, отриманого після шістнадцяти кроків основного циклу. Ця перестановка зворотна початковій перестановці.

Після виконання всіх кроків, блок даних вважається повністю зашифрованим і можна переходити до шифрування наступного блоку повідомлення.

1.4 Симетричний шифр блочного типу для шифрування текстової інформації

Advanced Encryption Standard (AES) являє собою симетричний шифр блочного типу, обраний урядом США для захисту секретної інформації і реалізований в програмному та апаратному забезпеченні в усьому світі для шифрування конфіденційних даних.

Національний інститут стандартів і технологій (NIST) почав розробку AES в 1997 році, коли оголосив про необхідність використання алгоритму наступника для застарілого алгоритму Data Encryption Standard (DES), який став вразливим для брут-форс атак.

В якості розширеного стандарту AES був обраний алгоритм Rijndael, розроблений бельгійськими криптографами Вінсентом Рейменом і Йоаном Дайменом і відрізнявся підвищеною безпекою, продуктивністю і гнучкістю.

Алгоритм Rijndael являє собою симетричний блочний шифр, який підтримує розміри ключів 128, 192 і 256 біт, причому дані обробляються в 128-бітних блоках, однак, крім критеріїв проектування AES, розміри блоків можуть бути дзеркальними для ключів. Rijndael використовує змінну кількість раундів, в залежності від розміру ключа і блоку, наступним чином:

- 9 раундів, якщо розмір ключа і блоку становить 128 біт;
- 11 раундів, якщо розмір ключа і блоку становить 192 біта;
- 13 раундів, якщо розмір ключа і блоку становить 256 біт.

Rijndael-це шифр лінійного перетворення підстановки, що не вимагає мережі Фейстеля. Він використовує потрійні стримані оборотні рівномірні перетворення (шари). Зокрема, це:

- Лінійне перетворення;
- Нелінійне перетворення;
- Перетворення ключа.

Ще до першого раунду виконується простий рівень додавання ключа, що додає безпеки. Після цього-раунди LN-1, а потім фінальний раунд. Перетворення утворюють форму при запуску, але до завершення всього процесу.

Форму можна розглядати як масив, структурований з 4 рядками, а номер стовпця-довжина блоку, поділена на довжину біт (наприклад, поділена на 32).

Ключ шифрування аналогічним чином являє собою масив з 4 рядками, але довжина ключа ділиться на 32, щоб вказати кількість стовпців. Блоки можуть бути інтерпретовані як одновимірні масиви 4-байтових векторів.

Точні перетворення відбуваються наступним чином: субтрансформація байтів нелінійна і працює на кожному з байтів форми незалежно - оборотна таблиця підстановок складається з двох перетворень.

Трансформація зсуву бачить, що форма зміщена по змінним зміщенням. Значення зсуву зсуву залежать від довжини блоку форми.

Перетворення за допомогою функції, яка змішує дані всередині кожного стовпця форми, бачить, що стовпці форми беруть поліноміальні характеристики за значеннями поля Галуа (28), помножені на x4 + 1 (за модулем) з фіксованим многочленом. Нарешті, відбувається перетворення в форму за допомогою унікального ключа, який застосовується в кожному окремому раунді, і функції XOR:

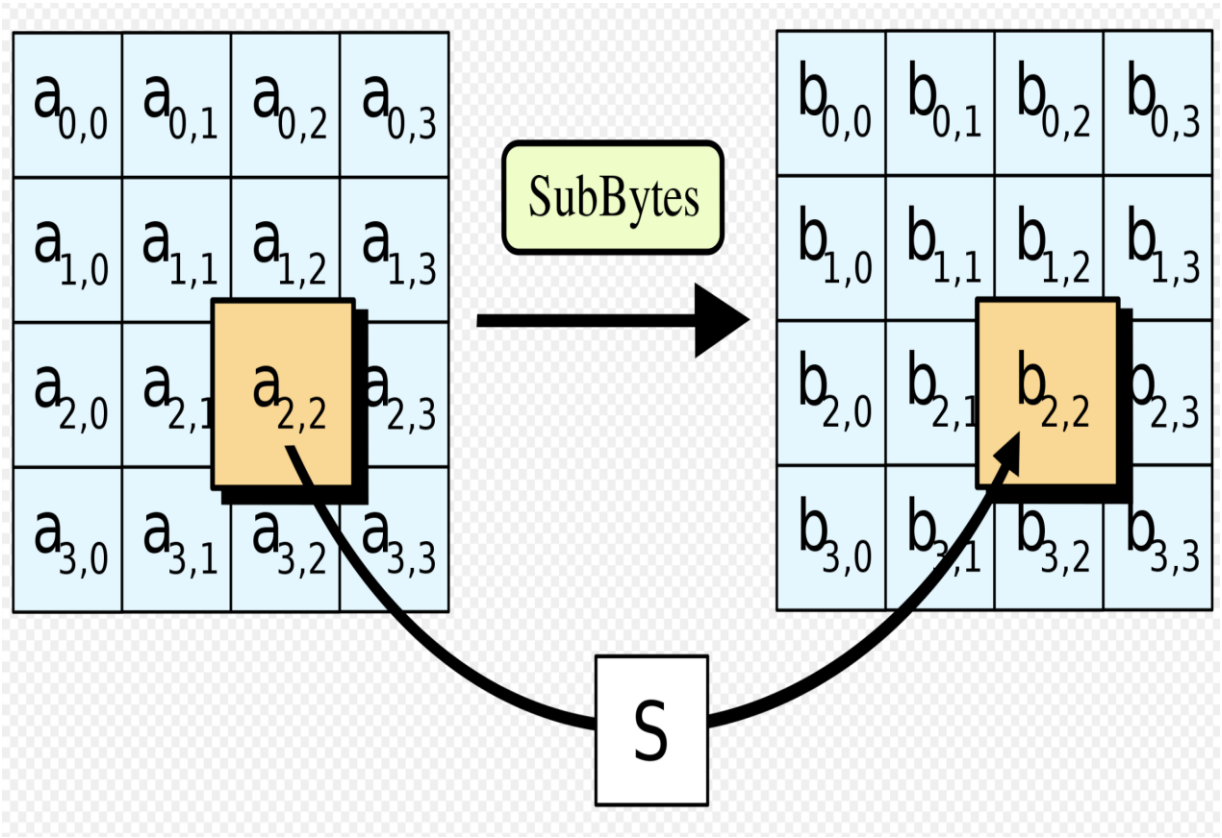


Рис. 1.3. - Схема симетричного шифрування за алгоритмом AES

Розклад ключів допомагає ключу шифрування визначати унікальні ключі за допомогою розширення ключа і вибір раунду.

В цілому, структура Rijndael демонструє високий ступінь модульної конструкції, яка повинна зробити модифікацію для протидії будь-якій атаці, навіть з урахуванням майбутніх технологій, набагато простіше, ніж з використанням застарілих алгоритмів, саме це робить дану структуру однією з найзручніших для використання в методах шифрування текстової інформації в мережі за допомогрю блокчейн технологій.

1.5 Алгоритм асиметричного шифрування в блокчейн

Асиметрична криптографія, також відома як криптографія з відкритим ключем, використовує загальнодоступні та закриті ключі для шифрування та дешифрування даних.

Ключі - це просто великі числа, які були з'єднані разом, але не ідентичні (асиметричні). Один ключ з пари може бути відомий всім, він називається відкритим ключем. Інший ключ з пари зберігається в секреті, він називається закритим ключем.

Будь-який з ключів може використовуватися для шифрування повідомлення, для дешифрування використовується протилежний ключ від того, який використовується для шифрування повідомлення.

Багато протоколів, таких як SSH, OpenPGP, S/MIME та SSL/TLS, базуються на асиметричній криптографії для функцій шифрування та цифрового підпису. Він також використовується в програмах, таких як браузери, яким необхідно встановити безпечне з'єднання по небезпечній мережі, наприклад, в інтернеті, або для перевірки цифрового підпису.

Надійність шифрування безпосередньо залежить від розміру ключа, а подвоєння довжини ключа забезпечує експоненціальне збільшення міцності, хоча і знижує продуктивність. У міру збільшення обчислювальної потужності і виявлення більш ефективних алгоритмів факторингу збільшується, і здатність збільшувати число також зростає.

При асиметричному шифруванні для забезпечення конфіденційності, цілісності, автентичності і відмовостійкості, користувачі і системи повинні бути впевнені, що відкритий ключ є справжнім, що він належить заявленій особі або суб'єкту і що він не був підроблений або замінений зловмисниками.

Не існує ідеального рішення проблеми аутентифікації з відкритим ключем. Найбільш поширеним підходом є інфраструктура відкритих ключів (PKI), в якій довірені сертифікаційні центри сертифікують права власності на пари ключів і сертифікати, але продукти шифрування, засновані на моделі Pretty Good Privacy

(включаючи OpenPGP), покладаються на децентралізовану модель аутентифікації, званої веб-службою довіри, яка спирається на індивідуальні схвалення зв'язку між Користувачем і відкритим ключем.

Вітфілд Діффі та Мартін Хеллман, дослідники зі Стенфордського університету, вперше публічно запропонували асиметричне шифрування у своїй статті 1977 року «нові напрямки в криптографії". За кілька років до Діффі і Хеллмана ця концепція була незалежно і таємно запропонована Джеймсом Еллісом, який працював у штаб-квартирі урядових комунікацій (GCHQ), британської розвідувальної та охоронної організації.

Асиметричний алгоритм, описаний в документі Діффі-Хеллмана, використовує спеціальні числа, для створення ключів дешифрування.

RSA (Rivest-Shamir-Adleman), найбільш широко використовуваний асиметричний алгоритм, вбудований в протокол SSL/TLS, який використовується для забезпечення безпеки зв'язку по комп'ютерній мережі.

RSA отримує свою безпеку від обчислювальної складності факторизації великих цілих чисел, які є добутком двох великих простих чисел. Множення двох великих простих чисел легко, але складність визначення вихідних чисел з сумарного факторингу - є основою безпеки криптографії з відкритим ключем.

Час, що витрачається на фактор продукту двох досить великих простих чисел, вважається занадто великим для основної частини атакуючих, за винятком національних державних суб'єктів, які можуть мати доступ до достатньої обчислювальної потужності.

RSA-ключі зазвичай мають довжину 1024 або 2048 біт, але експерти вважають, що в найближчому майбутньому можуть бути зламані 1024-бітові ключі, тому уряд і індустрія переходять на мінімальну довжину ключа 2048 біт.

Еліптична крива криптографії (ECC) завойовує популярність у багатьох експертів з безпеки в якості альтернативи RSA для реалізації криптографії з відкритим ключем. ECC - це метод шифрування з відкритим ключем, заснований на теорії еліптичних кривих, який може створювати більш швидкі, більш дрібні і більш ефективні криптографічні ключі.

ECC генерує ключі через властивості рівняння еліптичної кривої. Щоб зламати ECC, потрібно обчислити дискретний логарифм еліптичної кривої, і виявляється, що це значно складніше завдання, ніж факторинг.

Як результат, розміри ключів ECC можуть бути значно меншими, ніж необхідні RSA, але забезпечують еквівалентну безпеку з меншою обчислювальною потужністю та споживанням ресурсів акумулятора, що робить його більш придатним для мобільних додатків, ніж RSA.

Цифрові підписи засновані на асиметричній криптографії і можуть надавати запевнення щодо походження, ідентифікації та статусу електронного документа, транзакції або повідомлення, а також підтвердження інформованої згоди підписав.

Для створення цифрового підпису програмне забезпечення підпису (наприклад, програма електронної пошти) створює односторонній хеш електронних даних, які повинні бути підписані. Закритий ключ користувача використовується для шифрування хешу, повертаючи значення, унікальне для хешованих даних.

Зашифрований хеш поряд з іншою інформацією, такою як алгоритм хешування, формує цифровий підпис. Будь-яка зміна даних навіть в одному біті призводить до іншого значення хеш-функції.

Цей атрибут дозволяє іншим перевіряти цілісність даних, використовуючи відкритий ключ підписує особи для дешифрування хешу. Якщо дешифрований хеш відповідає другому обчисленому хешу тих же даних, він доводить, що дані не змінилися з моменту його підписання.

Якщо ці два хеші не збігаються, або дані якимось чином підроблені (що вказує на відмову цілісності), або підпис був створений за допомогою закритого ключа, який не відповідає відкритому ключу, представленому підписує особою (із зазначенням відмови аутентифікації).

Тож, для наглядного прикладу в використанні асиметричного шифрування потрібно представити схему шифрування, яку стане можливо використовувати при плануваннні шифрування текстової інформації за допомогою блокчейн технологій,

задля зручної та безпечної передачі даних в мережі, яка буде включати в себе публічний ключ, різні ключі, секретний ключ:

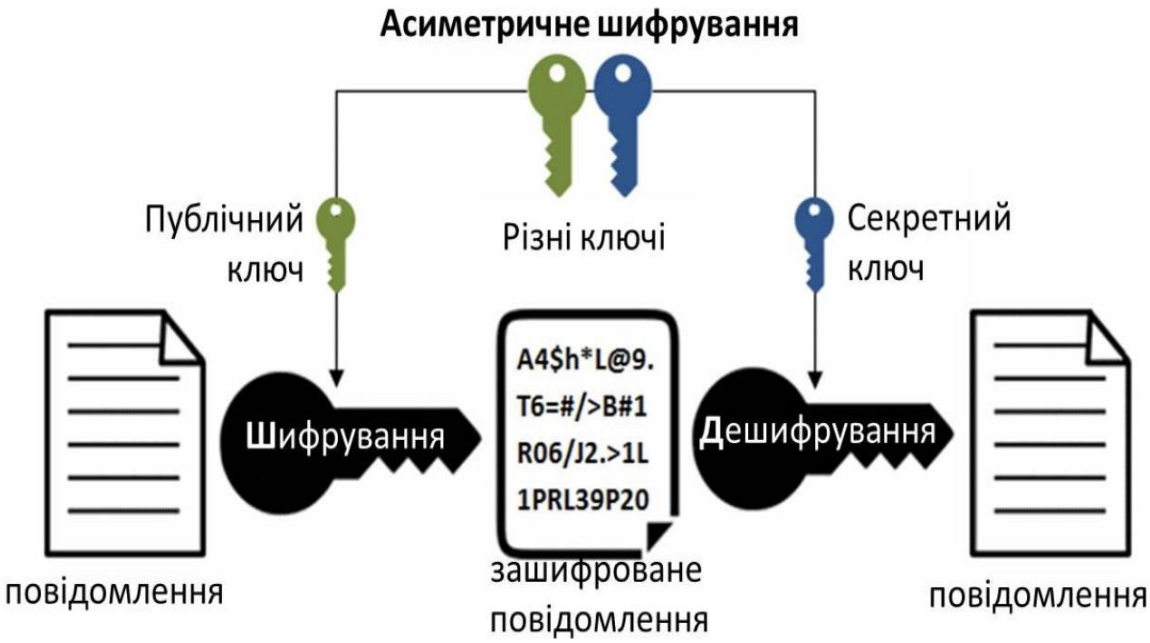


Рис. 1.4. - схема асиметричного шифрування

Цифровий підпис також перешкоджає підписаній стороні відмовитися від того, що вона щось підписала (властивість відмовостійкості). Якщо підписана сторона заперечує дійсний цифровий підпис, їх секретний ключ або був скомпрометований, або вони брешуть. У багатьох країнах, цифрові підписи мають однакову юридичну силу з більш традиційними формами підписів.

1.6 Шифрування текстової інформації в блокчейн за допомогою протоколу Діффі-Хеллмана

Перша публікація даного алгоритму з'явилася в сімдесятих роках XX століття в статті Вітфілда Діффі і Мартіна Хеллмана, в якій вводилися основні поняття криптографії з відкритим ключем.

Алгоритм Діффі-Хеллмана не застосовується для шифрування даних або формування електронного підпису. Його призначення - в розподілі ключів. Він

дозволяє двом або більше користувачам обмінятися без посередників ключем, який може бути використаний для симетричного шифрування.

Це була перша криптосистема, яка дозволяла захищати інформацію без необхідності використання секретних ключів, що передаються по захищених каналах.

Схема відкритого розподілу ключів, запропонована Діффі і Хеллманом, зробила справжню революцію в шифруванні, так як вирішувала основну проблему класичної криптографії - проблему розподілу ключів.

Алгоритм заснований на труднощі обчислення дискретних логарифмів. У цьому алгоритмі, як і в багатьох інших алгоритмах з відкритим ключем, обчислення виконуються по модулю деякого великого простого числа P. Спочатку спеціальним чином підбирається деяке натуральне число a, менше P. Якщо потрібно зашифрувати значення X, то обчислюється Y (1.1):

$$Y = Ax \bmod P. \quad (1.1)$$

При цьому, маючи X, обчислити Y легко. Зворотна задача обчислення X з Y є досить складною. Експонента X і називається дискретним логарифмом Y. Таким чином, знаючи про складність обчислення дискретного логарифма, число Y можна відкрито передавати навіть по незахищеному каналу зв'язку, так як при великому модулі P вихідне значення X підібрати буде практично неможливо. Алгоритм Діффі-Хеллмана для формування ключа заснований на цьому математичному факторі.

Нехай два користувача, користувач 1 і користувач 2, бажають сформувати загальний ключ для алгоритму симетричного шифрування. Спочатку вони повинні вибрати велике просте число P і деяке спеціальне число a, яке $1 < N < R-1$, таке, що всі числа з інтервалу [I, II, ..., PI] можуть бути представлені як різні ступені a Mod P.

Всім абонентам системи ці числа повинні бути відомі і можуть вибиратися відкрито. Це будуть загальні параметри.

Потім користувач 1 вибирає число X1, таке, що $X1<P$, яке потрібно формувати за допомогою генератора випадкових чисел. Це буде закритий ключ

першого користувача, і він повинен зберігатися в секреті. На основі закритого ключа користувач 1 обчислює число (1.2):

$$Y^1 = A^{x1} \, mod \, P \tag{1.2}$$

яке він відправляє другому користувачеві. Другий абонент надходить аналогічно, генеруючи X2 і обчислюючи:

$$Y^2 = A^{x2} \, mod \, P \tag{1.3}$$

Цей результат надсилається першому користувачеві. Після цього у користувачів є наступна інформація:

Таблиця 1.1 - Параметри ключів шифрування

	Загальні параметри	Відкритий ключ	Закритий ключ
1й користувач	P, A	Y_1	X_1
2й користувач		Y_2	X_2

З чисел Y1 і Y2, а також з особистих закритих ключів кожен користувач може згенерувати загальний секретний ключ Z для сеансу симетричного шифрування. Перший користувач:

$$Z = (Y^2)^{x1} \, mod \, P \tag{1.4}$$

Ніхто, крім першого користувача, не може цього зробити, так як число X1 таємно. Другий користувач може отримати таке ж число Z, використовуючи свої закритий ключ і відкритий ключ користувача 1:

$$Z = (Y^1)^{x2} \, mod \, P \tag{1.5}$$

Якщо весь протокол формування загального секретного ключа виконаний вірно, значення Z у одного і другого абонента повинні вийти однаковими. Причому, що найважливіше, зловмисник, не знаючи секретних чисел X1 і X2, не зможе обчислити Z.

Не знаючи X1 і X2, він може спробувати обчислити Z, використовуючи тільки передані відкрито значення P, A, Y1 і Y2.

Безпека формування загального ключа в алгоритмі Діффі-Хеллмана виходить з того факту, що, хоча відносно легко вирахувати експоненти по модулю простого числа, дуже важко обчислити дискретні логарифми. Завдання вважається нерозв'язним для великих простих чисел розміром сотні і тисячі біт, так як вимагає колосальних витрат обчислювальних ресурсів.

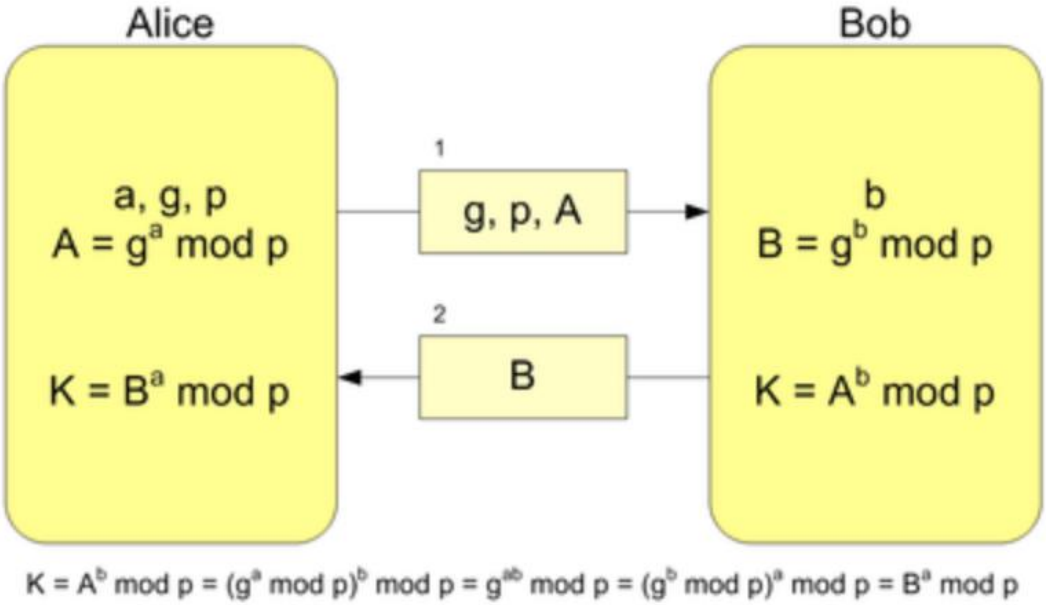


Рис. 1.5. - Схема протоколу Діффі-Хеллмана

Перший і другий користувачі можуть використовувати число Z як секретний ключ як для шифрування, так і для розшифрування даних. Таким же чином будь-яка пара абонентів може обчислити секретний ключ, відомий тільки їм.

1.7 Алгоритм з відкритим ключем RSA для шифрування текстової інформації

RSA - криптографічний алгоритм з відкритим ключем, що ґрунтується на обчислювальній складності задачі факторизації великих цілих чисел. RSA є першим алгоритмом шифрування з відкритим ключем.

Назва системи походить від перших букв прізвищ її авторів – Рональд Рівест, Аді Шамір і Леонард Адлеман – трьох вчених з Массачусетського технологічного інституту.

Після вивчення опублікованої в 1976 році статті Вітфілда Діффі та Мартіна Хеллмана "нові напрямки в криптографії", яка заклала основи криптографії з відкритим ключем, Рівест, Шамір і Адлеман приступили до пошуків математичної функції, яка дозволяла б реалізувати модель системи, описаної в статті.

Після роботи над більш ніж 40 можливими варіантами, вченим вдалося виявити алгоритм, що грунтується на тому, наскільки легко знаходити великі прості числа і наскільки складно розкладати на множники твір двох великих простих чисел.

Для шифрування використовується проста операція зведення в ступінь по модулю N. Для розшифрування необхідно обчислити функцію Ейлера від числа N, для цього необхідно знати розкладання числа N на прості множники (в цьому полягає завдання факторизації). У криптографічній системі RSA відкритий і закритий ключі складаються з пари цілих чисел.

Закритий ключ зберігається в секреті, а відкритий повідомляється іншому учаснику, або десь публікується.

Система базується на наступних фактах:

1. При відомих числах b і d обчислення числа a з порівняння (1.6)

$$2.\ \ a = b^d \ mod\ n \tag{1.6}$$

3. За складеним модулем n - це просте завдання;
4. Обчислення невідомого числа b при відомих числах d і a з порівняння по складеному модулю n є важким завданням;
5. Якщо відомо, що p і q прості числа і n = pq, то обчислити N легко, а знайти розкладання n на прості множники важко;
6. Якщо відомо розкладання n = pq на прості множники, то задача обчислення числа b з рівняння(1.7)

$$A = b^d \ mod\ n \tag{1.7}$$

може бути виконана.

Теоретичною основою криптосистеми RSA є теорема Ейлера: для будь-яких натуральних і взаємно простих чисел n і a справедливо рівність (1.8)

$$\mathrm{A}^{\varphi n} = 1 \ mod\ n \tag{1.8}$$

Тут φ(n) є функція Ейлера - кількість взаємно простих з n натуральних чисел від 1 до n.

З теорії чисел відомо, що якщо p і q прості числа, а n = pq, то

$$\varphi\, n = p - 1\; q - 1 \tag{1.9}$$

Крім того, з теореми Ейлера випливає, що якщо деяке число e взаємно просто з φ(n) , то рівняння (1.10)

$$de = 1\; mod\; \varphi\, n \tag{1.10}$$

або інакше (1.11)

$$de = k\; \varphi\, n + 1 \tag{1.11}$$

Однозначно вирішується щодо D. Рішення легко визначається розширеним алгоритмом Евкліда.

$$de = 1\; mod\; \varphi\, n \tag{1.12}$$

а x-передана інформація, то справедливо рівність (1.13)

$$(x^e)^d\; mod\; n = x^{k\varphi n+1}\; mod\; n = x^{\varphi nk}\; x\; mod\; n = x \tag{1.13}$$

Отже, за рівністю можна представити наступну схему:

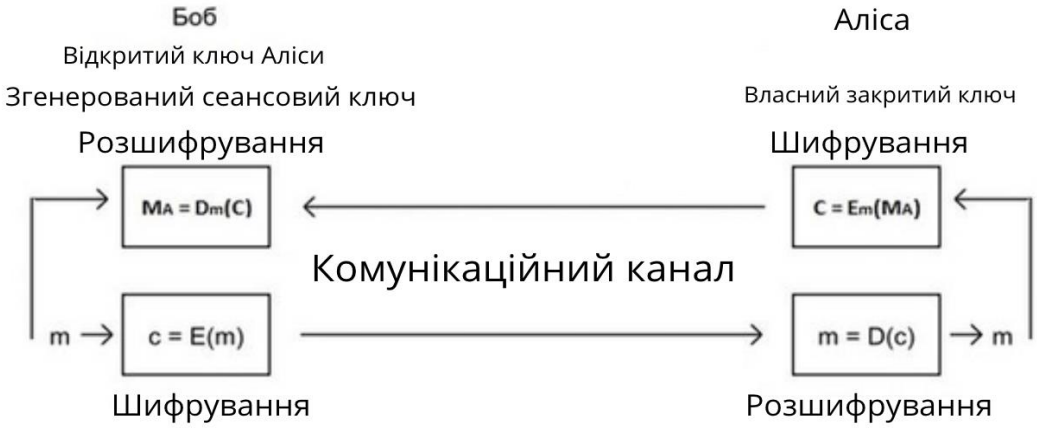


Рис. 1.6. - Схема алгоритму з відкритим ключем RSA

Фактично, останнє співвідношення є основою для формулювання системи RSA, яка відіграє значну роль в формуванні методів блокчейн шифрування даних користувачів в мережі.

1.8 Принцип роботи системи RSA для шифрування інформації задопомогою блокчейн технологій

Спочатку відбувається генерація пари ключів - відкритого і закритого. Генерація відбувається наступним способом:

1. Обирається два простих великих числа p і q, при цьому вони не рівні.
2. Обчислюється модуль числа:

$$3.\ \mathrm{N} = p * q \tag{1.14}$$

4. Обчислюється значення функції Ейлера від модуля числа N:

$$5.\ \varphi\,\mathrm{N} = p - 1 * q - 1 \tag{1.15}$$

6. Обирається деяке число e-відкрита експонента - яке лежить в інтервалі 1 < q < φ N і є взаємно простим зі значенням функції φ N.
7. Обчислюється число d - таємна експонента. До того ж, воно є мультиплікативним зворотнім до числа e по модулю φ N:

$$d * e\ = 1\ mod\ \varphi\,\mathrm{N} \tag{1.16}$$

В результаті виходить пара ключів: (e, N) – відкритий ключ і (d, N) – закритий ключ.

Користувач А і користувач В обмінюються повідомленнями в Інтернеті. Щоб підтримувати листування в секреті, вони використовують шифрування. Користувач В заздалегідь згенерував пару ключів, а потім передав відкритий ключ користувачеві А, який відправляє зашифроване повідомлення.

Шифрування: Користувач А шифрує повідомлення m за допомогою відкритого ключа другого користувача (e, N) і відправляє його:

$$c * e\ m\ = m^n\ (\mathrm{N}) \tag{1.17}$$

Розшифрування: прийнявши зашифроване повідомлення, Користувач В розшифровує його, використовуючи закритий ключ (d, N):

$$m * n = d\ c\ = c^d\ mod(\mathrm{N}) \tag{1.18}$$

Виходячи з формул, потрібна наглядна демонстрація принципу роботи алгоритму RSA, який буде використано для роботи з шифруваня текстової інформації.

А отже, з даних формул випливає схема роботи алгоритму RSA:

Рис. 1.7. - Принцип роботи RSA

Зі схеми принципу роботи алгоритму RSA стає зрозумілим цілісність його використання при роботі з шифрування текстової інформації за допомогою блокчейн технологій.

1.9 Приклад шифрування та розшифрування алгоритму з відкритим ключем RSA

Потрібно зашифрувати повідомлення "RSA". Позначимо кожну букву їх порядковими номерами в англійському алфавіті R – 18, S – 19, A – 1.

Далі слідуємо алгоритму:

1. Вибираємо прості числа (для простоти обчислень візьмемо невеликі): p = 3, q = 11.
2. Обчислюємо модуль N:

$$N = p * q = c^d \ mod(\text{N}) \tag{1.19}$$

3. Знаходимо функцію Ейлера від модуля числа N:

$$\varphi\, N = 3 - 1 * 11 - 1 = 2 - 10 = 20 \tag{1.20}$$

4. Обираємо відкриту експоненту: e = 7.

5. Обчислюємо відкриту експоненту:

$$d * 7 = 1\ mod(20), d = 3 \tag{1.21}$$

Отриманим відкритим ключем (7,33) шифруємо кожну літеру вихідного повідомлення:

$$\begin{aligned} \text{c1} &= 18^7\ mod\ 33 = 6; \\ \text{c1} &= 18^7\ mod\ 33 = 13; \\ \text{c1} &= 18^7\ mod\ 33 = 1. \end{aligned} \tag{1.22}$$

Завдяки даним рівнянням, стає зрозумілим принцип шифрування RSA:

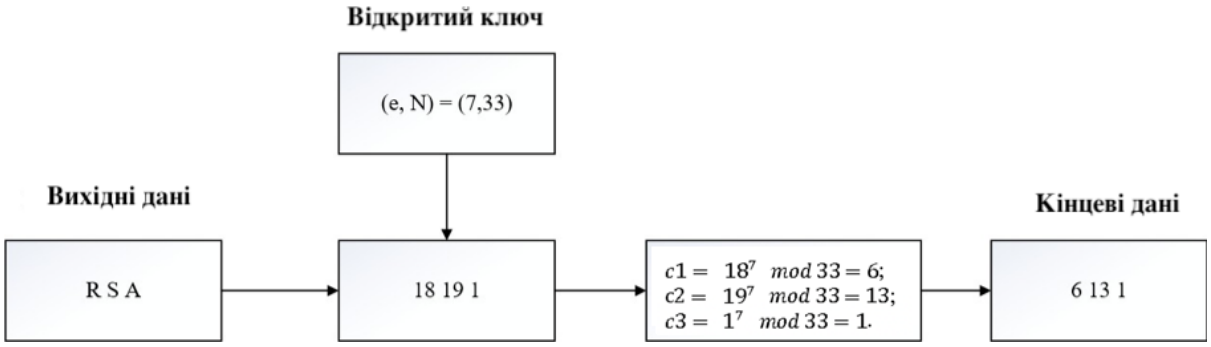


Рис. 1.8. - Приклад шифрування RSA

Щоб розшифрувати отримане повідомлення, використовуємо закритий ключ (3, 33):

$$\begin{aligned} \text{c1} &= 6^7\ mod\ 33 = 18; \\ \text{c1} &= 13^3\ mod\ 33 = 19; \\ \text{c1} &= 1^7\ mod\ 33 = 1. \end{aligned} \tag{1.23}$$

Отримуємо вихідне повідомлення.

1.10 Приклади месенджерів побудованих на основі блокчейн технологій

На даний час, з стрімким розвитком блокчейн технологій, активно стали з’являтись мессенджери на офнові блокчейн, основною задачею яких стає захищена передача даних, файлів, зоображень, аудіо-файлів в мережі. Одними з перших почали створювати захищені месенджери:

1. Платформа Dust стала одним з перших на ринку чат додатків на основі блокчейна. Розроблений в Лос-Анджелесі месенджер з шифруванням даних був випущений 1 березня 2014 року і набув широкого розголосу в ЗМІ завдяки одному з головних прихильників платформи, Марку Кьюбану. Dust пропонує користувачам кілька унікальних функцій - наприклад, після 24 годин повідомлення автоматично видаляються з телефонів відправника і одержувача (звідси і назва програми, на англ. Dust означає «прати, чистити»). Крім того, ви можете налаштувати свої повідомлення таким чином, щоб вони зникали після сповіщення про прочитання.
2. Cryptviser Cryptviser - це децентралізована платформа, яка усуває загрозу атак по типу «людина посередині», Man in The Middle, MITH (вид атаки в криптографії, коли зловмисник таємно ретранслює і при необхідності змінює зв'язок між двома сторонами, які вважають, що вони безпосередньо спілкуються один з одним, - прим. ред.). Більшість месенджерів зберігають ваші дані в централізовану базу. Ті, хто має доступ до інформації, можуть легко переглядати, блокувати і контролювати такі системи. Cryptvisor дозволяє уникнути цих ризиків, оскільки із загального рівняння зникають централізовані сервери і призначена для користувача інформація. В інтерв'ю журналу App Developer Magazine директор з комунікацій Марк Беббіт підкреслив, що мета його компанії - забезпечити рядовим користувачам рівень безпеки не нижче прийнятого для корпорацій і урядів по всьому світу.
3. Мессенджер Echo використовує блокчейн-технологію Graphene, щоб забезпечити конфіденційний і безпечний обмін повідомленнями. Через нього можна відправляти мультимедійні повідомлення з шифруванням всіх аудіо- і відеопотоків в режимі реального часу. Платформа децентралізована і дозволяє користувачам відправляти платежі в криптовалюта безпосередньо з програми. У недавньому інтерв'ю генеральний директор компанії Крістоф Герінг розповів, що Echo використовує спеціально розроблений протокол IPFS (від англ. InterPlanetary File System - міжпланетна файлова система) для прискореного обміну повідомленнями. IPFS спирається на адресну систему, а не на

стандартний протокол HTTP. Як правило, в месенджерах на блокчейне потрібно, щоб одержувач і відправник спілкувалися безпосередньо на блокчейне, що може вимагати наявності великих ресурсів. Echo усуває цю потребу за рахунок клієнта IPFS.

4. Додаток e-Chat вводить в гру IPFS в поєднанні з P2P-технологією обміну повідомленнями. На відміну від інших традиційних платформ обміну повідомленнями, тут відсутній центральний вузол зберігання ваших даних. Децентралізація усуває проблеми, пов'язані з атаками по типу «людина посередині». Платформа також дозволяє користувачам винагороджувати блогерів за їх контент. Ця функція називається Cryptolike, і вона входить в розділ створення контенту платформи, де користувачі можуть створювати канали підписки. Платформа e-chat включає в себе вбудовану платіжну систему для безконтактних платежів NFC і QR. Таким чином, ви можете переводити традиційні гроші або криптовалюта в особистих повідомленнях. У платформі також є вбудоване ПО для конвертації криптовалюта, що дозволяє обмінювати її безпосередньо з гаманця вашого телефону. Вам надається безліч різних способів доступу до своєї інформації, включаючи сканування відбитка пальця, розпізнавання особи і особисті ключі доступу до гаманця.
5. Неймовірно популярний месенджер Telegram успішно зібрав понад 1,7 млрд доларів в цьому році на своєму первісному розміщенні монет (ICO). Telegram вже давно підтримує захищений алгоритмамишифрування обмін повідомленнями, проте тепер компанія повністю переходить на інтеграцію криптографічних рішень. Перша інтеграція такого роду почалася в цьому місяці в рамках нової програми для ідентифікації Passport. Passport дозволяє Telegram зберігати ваші найважливіші і особисті документи в захищеній блокчейн-мережі. Компанія використовує наскрізне шифрування, а співробітники компанії підкреслюють, що у них немає доступу до ваших даних.
6. В Status для захисту інформації своїх користувачів від сторонніх очей використовується добре пророблений блокчейн Ethereum. Платформа пропонує наскрізне шифрування всіх повідомлень. Крім того, ви зможете переглянути

велику добірку децентралізованих додатків (Dapps) через вбудований браузер платформи. Користувачі Status знайомі з багатьма унікальними функціями, включаючи децентралізований ринок праці, біржу, систему прогнозування і сервіс цифрової ідентифікації. Платформа також дозволяє творчим людям безпечно просувати свої роботи через блокчейн Ethereum

Висновок до розділу 1

Під час ознайомлення з методами шифрування текстової інформації, були виявлені найбільш ефективні методи для роботи з блокчейн технологіями. В результаті огляду було обрано методи які забезбечать найбільш ефективний та простий в роботі метод шифрування текстової інформації, яка буде передаватись між користувачами в мережі. Вибір ґрунтується на літературі провідних дослідників блокчейн технологій. Для перевірки результату потрібно буде виконати логін до сайту, з різних IP адрес, де буде знаходитись текстова інформація захищена шифруванням на базі блокчейн технологій.

2 МАТЕМАТИЧНА СКЛАДОВА ШИФРУВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ ЗА ДОПОМОГОЮ БЛОКЧЕЙН ТЕХНОЛОГІЙ

2.1 Математичні основи блокчейн технологій для шифрування текстової інформації

Сьогодні блокчейн технології продовжують набирати популярність, а індустрія розробляти все більш нові застосування для роботи з інформацією.

Однією з причин такої популярності є математична база, на якій будується блокчейн. Завдяки цьому система функціонує в умовах повної відсутності довіри між учасниками мережі, виключаючи дію людського чинника.

Фундаментальною частиною блокчейн є криптографічні алгоритми. Зокрема, алгоритм ECDSA - Elliptic Curve Digital Signature Algorithm, який використовує еліптичні криві (elliptic curve) і кінцеві поля (finite field) для підпису даних, щоб третя сторона могла підтвердити автентичність підпису, зробивши неможливою її підробку.

У ECDSA для підпису і верифікації використовуються різні процедури, що складаються з декількох арифметичних операцій.

Еліптична крива над полем K - це кубічна крива над замиканням алгебри поля K, що задається рівнянням третьої міри з коефіцієнтами з поля K і "точкою на нескінченності".

Використовуючи теорію еліптичних функцій, можна показати, що еліптичні криві, визначені над комплексними числами, відповідають вкладанням тору в складну проективну площину. Тор також є абелевою групою, і це відповідність також є груповим ізоморфізмом.

Еліптичні криві особливо важливі в теорії чисел і становлять основну область сучасних досліджень; наприклад, вони були використані в доказі Ендрю Вайлсом останньої теореми Ферма. Вони також знаходять застосування в криптографії еліптичних кривих (ECC) та цілочисельній факторизації.

Однією з форм еліптичних кривих є криві Вейерштраса. $y^2 = x^3 + ax + b$ Для коефіцієнтів a = 0 і b = 7, графік функції набирає наступного вигляду:

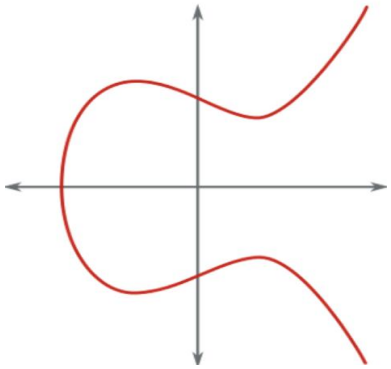


Рис. 2.1. - Базова еліптична крива для побудови блокчейн

Завдякий базовій еліптичній кривій, ми можемо краще зрозуміти матиматичну складову блокчейн.

2.2 Еліптична крива для побудови блоків в блокчейн

Еліптичні криві мають декілька цікавих властивостей, наприклад, не вертикальна лінія, що перетинає дві не дотичні точки на кривій. Сумою двох точок на кривій P + Q називається точка R, яка є відображенням точки, - R (побудованій шляхом продовження прямій (P; Q) до перетину з кривою) відносно осі X:

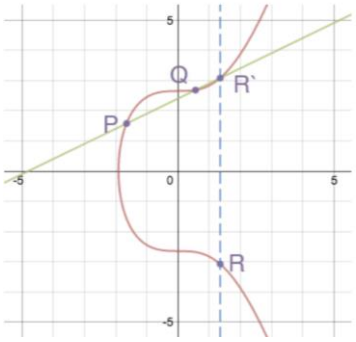


Рис. 2.2. - еліпптична крива з сумою двох точок на кривій

Якщо ж провести пряму через дві точки, що мають координати виду P (a, b) і Q (a, - b), то вона буде паралельна осі ординат.

В цьому випадку не буде третьої точки перетину. Щоб розв'язати цю проблему, вводиться так звана точка на нескінченності (point of infinity), що означає як O. Тому, якщо перетин відсутній, рівняння набирає наступного вигляду P + Q = O.

Якщо ми хочемо скласти точку саму з собою (подвоїти її), то в цьому випадку просто проводиться дотична до точки Q.

Отримана точка перетину відбивається симетрично відносно осі X.

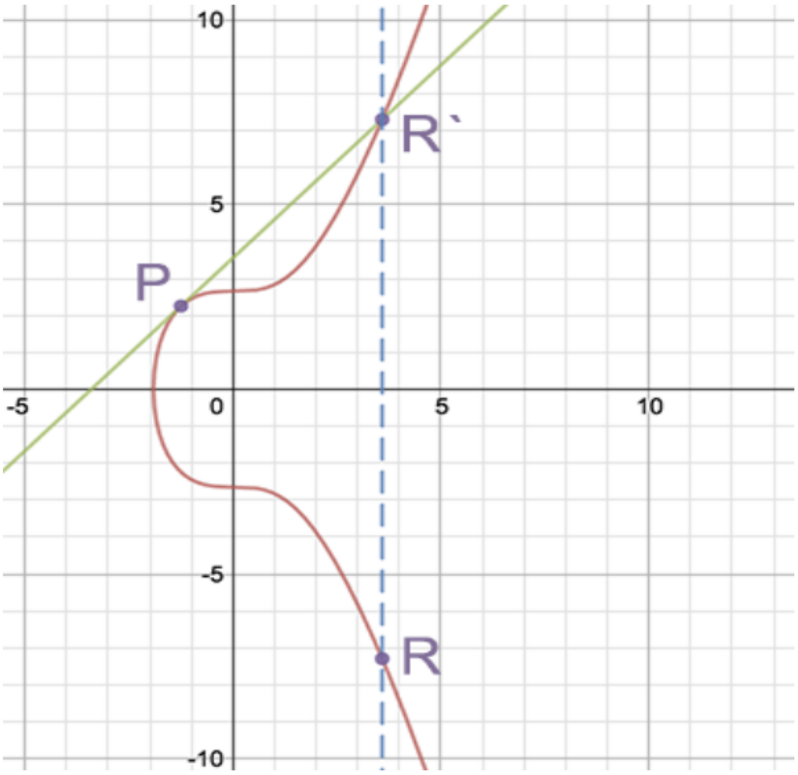


Рис. 2.3. - подвоення точок на еліптичній кривій

Ці операції дозволяють провести скалярне множення точки R = k*P, складаючи точку P саму з собою k разів. Проте відмітимо, що для роботи з великими числами використовуються швидші методи.

В еліптичній криптографії (ECC) використовується така ж крива, що тільки розглядається над деяким кінцевим полем. Кінцеве поле в контексті ECC можна представити як зумовлений набір позитивних чисел, в якому повинен опинятися результат кожного обчислення.

Наприклад, 9 mod 7 = 2. Тут ми маємо кінцеве поле від 0 до 6, і усі операції по модулю 7, над яким би числом вони не здійснювалися, дадуть результат, що потрапляє в цей діапазон.

Усі названі вище властивості (складання, множення, точка в нескінченності) для такої функції залишаються в силі, хоча графік цієї кривої не буде схожий на еліптичну криву.

Еліптична крива блокчейн, $y^2 = x^3 + 7$, визначена на кінцевому полі по модулю 67, виглядає таким чином:

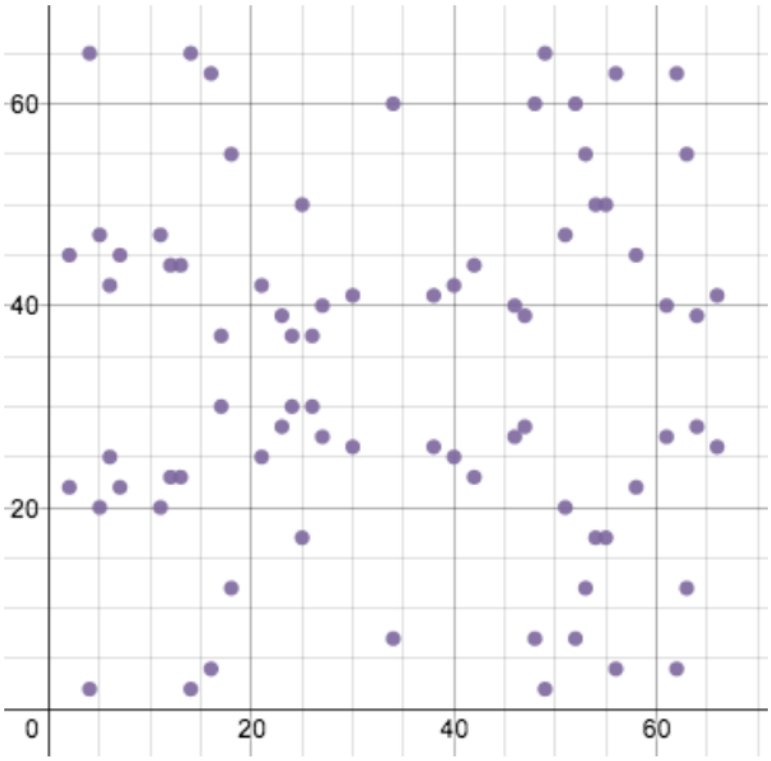


Рис 2.4. - Еліптична крива блокчейну визначена на кінцевому полі по модулю 67

Це безліч точок, в яких усі значення x і y є цілими числами між 0 і 66. Прямі лінії, намальовані на цьому графіку, тепер як би "обертатимуться" навколо поля, як тільки досягнуть бар'єру 67, і продовжаться з іншого його кінця, зберігаючи колишній нахил, але із зрушенням. Тож, для розуміння матиматичної складової блокчейн технологій, потрібно показати накладення точок на графіку, для розуміння повного плану.

Наприклад, складання точок (2, 22) і (6, 25) в цьому конкретному випадку виглядає так:

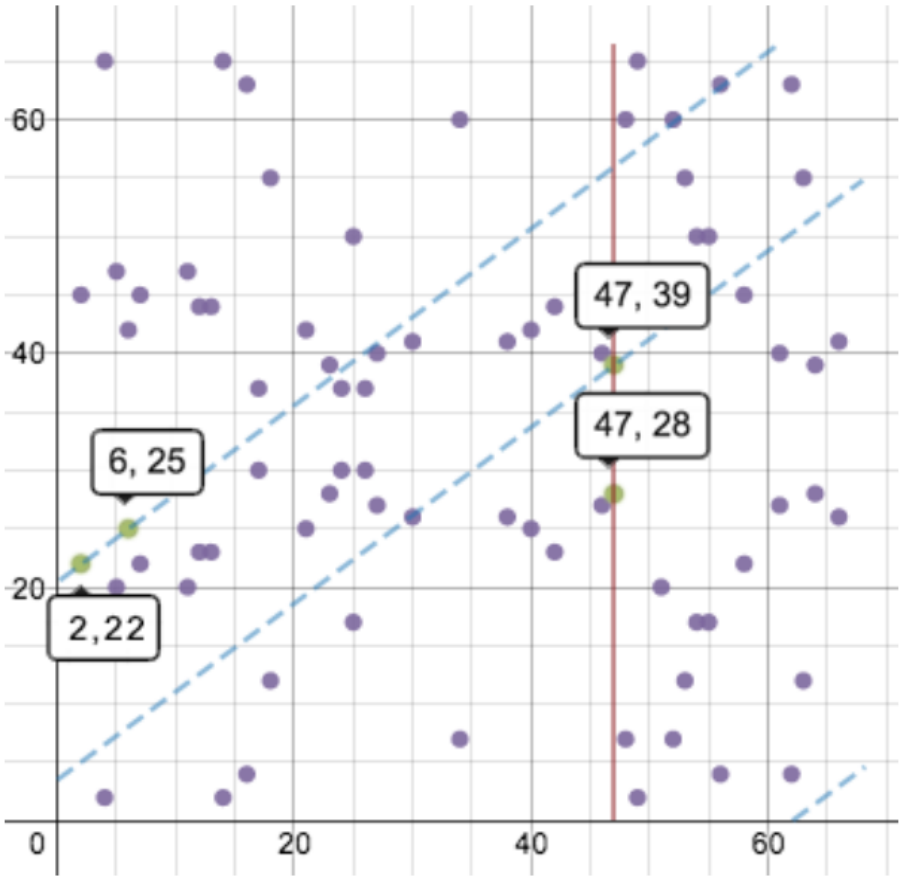


Рис 2.5. - Складення точок (2, 22) та (6, 25)

Такого роду еліптичні криві і кінцеві поля використовуються для підпису даних, щоб третя сторона могла підтвердити автентичність підпису, роблять неможливим її підробку.

2.3 ECDSA алгоритм для підбору ключів шифрування в блокчейн технології

В протоколі блокчейн зафіксований набір параметрів для еліптичної кривої і її кінцевого поля, щоб кожен користувач використав строго певний набір рівнянь яким надаються певні параметри.

Серед зафіксованих параметрів виділяють рівняння кривої (equation), значення модуля поля (prime modulo), базову точку на кривій (base point) і порядок базової точки (order).

Про обчислення порядку базової точки ви можете шанувати тут. Цей параметр підбирається спеціально і є дуже великим простим числом. У разі биткойна використовуються наступні значення:

Рівняння еліптичної кривої : $y^2 = x^3 + 7$ Простий модуль: 7134 - 123 - 12 - 11 - 10 - 9 - 8 - 1 = FCFEEFFF FEEFEFEF FCDEEEEF CDFFEFEFF FFCEFNFF FNEENFFF FENFDFEE FEELFFC2L

Базова точка:

05 21bh632o JSGTENCC 371623c5 LSdaHD21 enaHEVWs1 Ndsawu9 5sdasYxb 1are1ODs 846hqJD1 Ln21SFBD 76dha612c jd218dFs MS21G231 E2193173 97dhEwesq JLjds218

Жирним шрифтом виділена координата X в шістнадцятиричному записі. За нею відразу йде координата Y.

Порядок: FCFEEFFF FEEFEFEF FCDEEEEF CDFFEFEFF FFCEFNFF FNEENFFF FENFDFEE FEELFFC2L

Цей набір параметрів для еліптичної кривої відомий як secp256k1 і є частиною сімейства стандартів SEC (Standards for Efficient Cryptography), пропонованих для використання в криптографії. У блокчейні крива secp256k1 використовується спільно з алгоритмом цифрового підпису ECDSA (elliptic curve digital signature algorithm).

У ECDSA секретний ключ - це випадкове число між одиницею і значенням порядку. Відкритий ключ формується на підставі секретного: останній множиться на значення базової точки. Рівняння має наступний вигляд:

Відкритий ключ = секретний ключ * G

Це показує, що максимальна кількість секретних ключів (отже, біткоїн-адрес) - звичайно, і дорівнює порядку. Проте порядок є неймовірно великим числом, так що випадково або навмисно підібрати секретний ключ іншого користувача нереально. Обчислення відкритого ключа виконується за допомогою

тих же операцій подвоєння і складання точок. Це тривіальне завдання, яке звичайний персональний комп'ютер або смартфон вирішує за мілісекунди.

А ось зворотне завдання (отримання секретного ключа по публічному) - є проблемою дискретного логарифмування, яка вважається обчислювально складною (хоча строгого доказу цьому факту немає).

Кращі відомі алгоритми її рішення, на зразок ро Полларда, мають експоненціальну складність.

Для secp256k1, щоб вирішити завдання, треба близько 2^128 операцій, що зажадає часу обчислення на звичайному комп'ютері, порівнянного з часом існування Всесвіту.

Коли пара секретний/публічний ключ отримана, її можна використати для підпису даних. Ці дані можуть бути будь-якої довжини. Зазвичай першим кроком виконується хешування даних з метою набуття унікального значення з числом бітів, рівним битности порядку кривої (256).

Після хешування, алгоритм підпису даних z виглядає таким чином.

- Тут, G - базова точка, n - порядок, a d - секретний ключ.
- Вибирається деяке ціле k в межах від 1 до n - 1
- Розраховується точка (x, y) = k * G з використанням скалярного множення
- Знаходиться r = x mod n.
- Якщо r = 0, то повернення до кроку 1 Знаходиться s = (z + r * d) / k mod n.
- Якщо s = 0, то повернення до кроку 1 Отримана пара (r, s) є нашим підписом

Після отримання даних і підпису до них, третя сторона, знаючи публічний ключ, може їх верифицировать. Кроки для перевірки підпису такі (Q - відкритий ключ) : Перевірка, що і r, і s знаходяться в діапазоні від 1 до n - 1

- Розраховується w = s - 1 mod n
- Розраховується u = z * w mod n
- Розраховується v = r * w mod n
- Розраховується точка (x, y) = uG + vQ
- Якщо r = x mod n, то підпис вірний, інакше - недійсна

- Насправді, uG + vQ = u + vdG = (u + vd) G = (zs - 1 + rds - 1) G = (z + rd) s - 1g = kG останній рівність використовує визначення s на етапі створення підпис.

Безпека ECDSA пов'язана із складністю завдання пошуку секретного ключа, описаного вище. Крім цього, безпека початкової схеми залежить від "випадковості" вибору k при створенні підпису.

Якщо одно і те ж значення k використати більше одного разу, то з підписів можна витягнути секретний ключ, що і сталося з PlayStation 3.

Тому сучасні реалізації ECDSA, у тому числі використовувані у більшості биткойн-кошельков, генерують k детерміновано на основі секретного ключа і підписаного повідомлення.

Висновок до розділу 2

Було визначено метематичну основу для роботи з методами шифрування тексової інформації за допомогою блокчейн технологій. Еліптичні криві та ECDSA алгоритм для роботи з ключами наглядно показують як працювати з блоками в блокчейн для шифрування текстової інформації. На базі яких, буде успішно виконано захист текстової інформації що буде передаватись між користувачами в мережі.

3 РОЗРОБКА АРХІТЕКТУРИ БЛОКЧЕЙН ДЛЯ ШИФРУВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ

3.1 Логіка блокових елемантів для шифрування текстової інформації

Термін блокчейн вперше був описаний ще в 1991 році. Група дослідників хотіла створити інструмент для встановлення часових позначок цифрових документів, щоб їх не можна було відновити чи змінити. Далі техніку адаптував і винайшов Сатоші Накамото. У 2008 році Накамото створив першу криптовалюту, проект на основі блокчейна під назвою Bitcoin.

Загалом, технологія блокчейн має основні характеристики децентралізації, підзвітності та безпеки. Ця методика може підвищити експлуатаційну ефективність та значно заощадити витрати.

Попит та використання додатків, побудованих на архітектурі блокчейн, лише розвинуться. блокчейн - це ланцюг блоків, які містять конкретну інформацію (базу даних), але безпечним та справжнім способом, який об'єднується в мережу (одноранговий.

Іншими словами, блокчейн - це комбінація комп'ютерів, пов'язаних один з одним замість центрального сервера, тобто вся мережа децентралізована.

Щоб зробити це ще простішим, концепцію блокчейн можна порівняти з роботою, виконаною з Google Docs. Ви можете згадати дні перекидання документа. документи та очікування, коли інші учасники внесуть необхідні зміни.

У наші дні за допомогою Google Docs можна працювати над одним і тим же документом одночасно.

Техніка blockchain дозволяє поширювати цифрову інформацію, а не копіювати. Ця розподілена книга забезпечує прозорість, довіру та безпеку даних.

Архітектура блокчейн використовується дуже широко у фінансовій галузі. Однак у наші дні ця технологія використовується не тільки для криптовалют, але і для ведення діловодства, цифрових нотаріусів та розумних контрактів, які будуть основною складовою шифрування текстової інформації за допомогою блокчейн технологій.

Отже, необхідно продемонструвати модель:

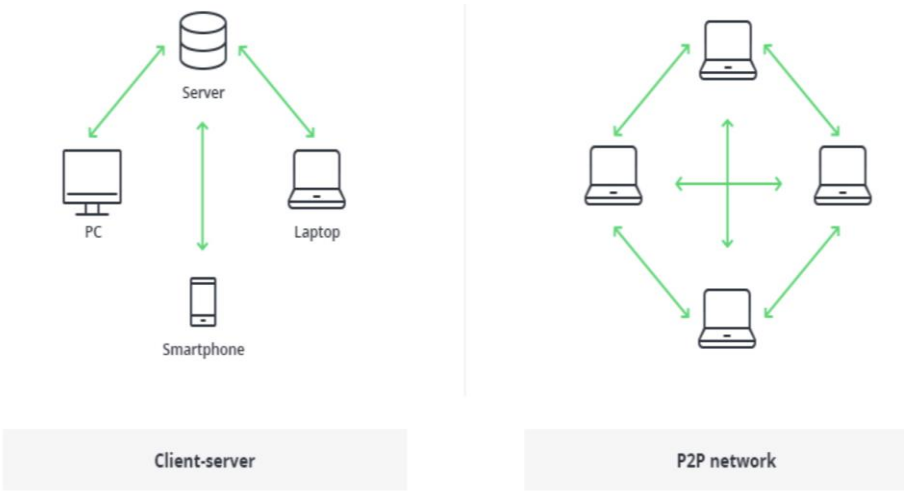


Рис. 3.1. - Схема роботи клієнт-серверної та P2P мереж

Традиційна архітектура всесвітньої павутини використовує мережу клієнт-сервер. У цьому випадку сервер зберігає всю необхідну інформацію в одному місці, щоб її було легко оновити, оскільки сервер є централізованою базою даних, керованою низкою адміністраторів з дозволами.

У випадку розподіленої мережі архітектури блокчейн кожен учасник мережі підтримує, затверджує та оновлює нові записи. Система контролюється не лише окремими особами, але й усіма в мережі блокчейн.

Кожен член гарантує, що всі записи та процедури в порядку, що призводить до дійсності та безпеки даних. Таким чином, сторони, які не обов'язково довіряють один одному, здатні досягти спільного консенсусу.

Для узагальнення речей блокчейн - це децентралізована, розподілена книга (державна чи приватна) різних видів транзакцій, розташованих у мережу P2P. Ця мережа складається з багатьох комп'ютерів, але таким чином, що дані не можуть бути змінені без консенсусу всієї мережі (кожен окремий комп'ютер).

Структура технології блокчейн представлена переліком блоків з транзакціями в певному порядку. Ці списки можна зберігати як плоский файл (txt. формат) або у вигляді простої бази даних. Дві життєво важливі структури даних, що використовуються в блокчейні, включають:

- Покажчики - змінні, які зберігають інформацію про розташування іншої змінної. Зокрема, це вказує на положення іншої змінної.
- Пов'язані списки - послідовність блоків, де кожен блок має конкретні дані та посилання на наступний блок за допомогою вказівника:

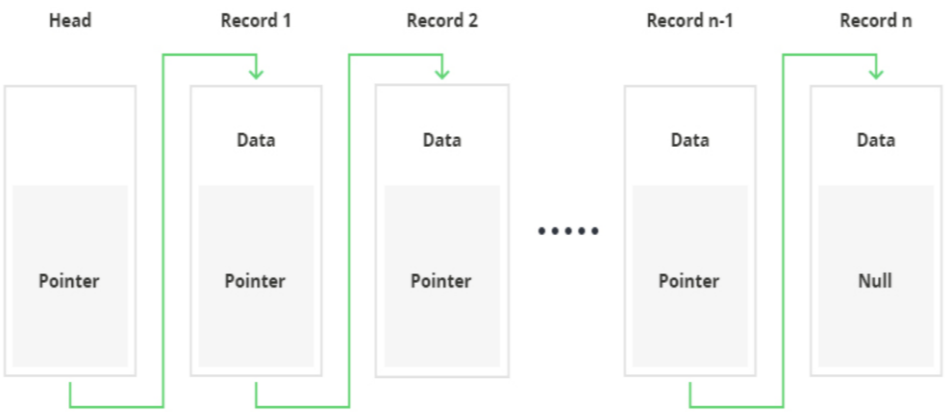


Рис. 3.2. - Хешування блоків в блокчейн

За логікою, перший блок не містить вказівника, оскільки цей перший у ланцюжку.

В основному, наступна діаграма послідовності блокчейна - це пов'язаний список записів:

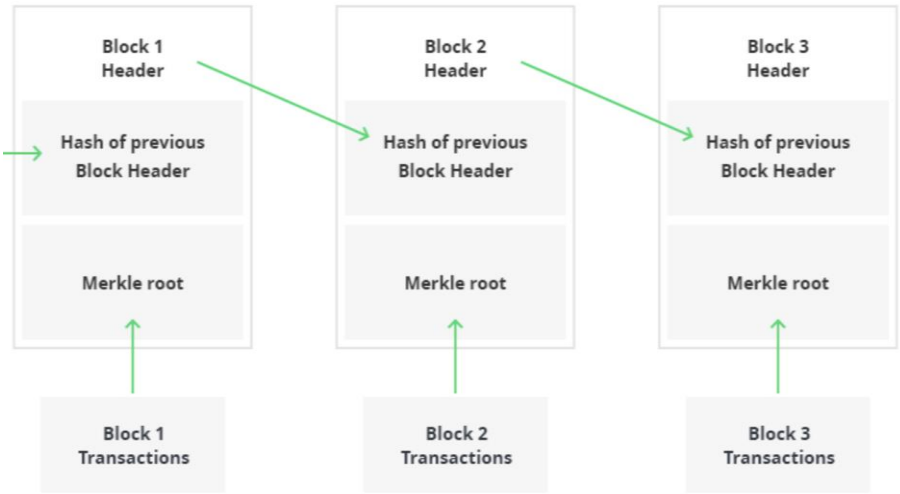


Рис. 3.3. - Структура будови блоків

Архітектура блокчейн може служити наступним цілям для організацій та підприємств:

Зниження витрат - багато грошей витрачається на підтримку баз даних, що знаходяться в центрі (наприклад, банки, урядові установи), зберігаючи дані, захищені від кіберзлочин та інших корупційних намірів.

Історія даних - у структурі блокчейн можна перевірити історію будь-якої транзакції в будь-який момент часу. Це постійно зростаючий архів, тоді як централізована база даних - це більше знімок інформації в певній точці.

Дійсність даних та безпека - після введення даних важко підробити через характер блокчейна. Потрібен час, щоб продовжити перевірку записів, оскільки процес відбувається в кожній незалежній мережі, а не через складну потужність обробки. Це означає, що система жертвує швидкістю продуктивності, але натомість гарантує високу безпеку та обґрунтованість даних.

Типи архітектури блокчейн:

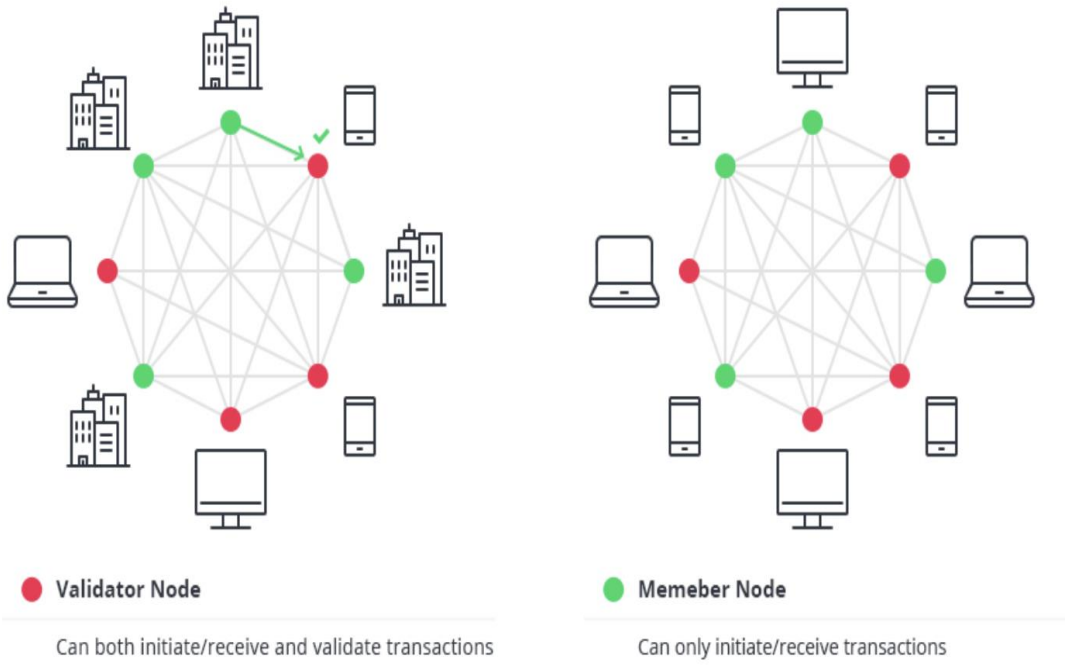


Рис. 3.4. - Вузли в публічних та приватних блокчейн

Усі структури блокчейна поділяються на три категорії:

- Публічна архітектура блокчейна
- Публічна передача блоків
- Публічна передача вузлів

- Публічна архітектура blockchain означає, що дані та доступ до системи доступні кожному, хто бажає брати участь (наприклад, системи Bitcoin, Ethereum та Litecoin блокчейн є загальнодоступними.
- Приватна архітектура блокчейн

На відміну від публічної архітектури блокчейн, приватна система контролюється лише користувачами певної організації або уповноваженими користувачами, які мають запрошення на участь.

- Консорціум блокчейн архітектури

Ця блокчейн-структура може складатися з кількох організацій. У консорціумі процедури встановлюються та контролюються попередньо призначеними користувачами.

Наступна таблиця дає детальне порівняння між цими трьома блокчейн-системами:

Property	Public blockchain	Consortium blockchain	Private blockchain
Consensus determination	All miners	Selected set of nodes	Within one organization
Read permission	Public	Public or restricted	Public or restricted
Immutability level	Almost impossible to tamper	Could be tampered	Could be tampered
Efficiency (use of resources)	Low	High	High
Centralization	No	Partial	Yes
Consensus process	Permissionless	Needs permission	Needs permission

Рис. 3.5. - Таблиця порівняння блокчейн систем

Як уже згадувалося, блокчейн - це розповсюджений журнал, де всі сторони мають локальну копію. Однак, виходячи з типу структури блокчейна та її контексту, система може бути більш централізованою або децентралізованою. Це просто стосується дизайну архітектури блокчейн і хто керує книгою.

Приватний блокчейн вважається більш централізованим, оскільки він контролюється певною групою з підвищеною конфіденційністю. Навпаки, публічний блокчейн відкритий і таким чином децентралізований.

У публічному блокчейні всі записи видно громадськості, і кожен може взяти участь у процесі угоди. З іншого боку, це менш ефективно, оскільки потрібно багато часу, щоб прийняти кожен новий запис в архітектуру блокчейн.

Що стосується ефективності, час для кожної транзакції в публічному блокчейні є менш екологічним, оскільки вимагає величезної кількості обчислювальної потужності порівняно з приватною архітектурою блокчейна.

3.2 Основні компоненти архітектури блокчейн для шифрування інформації

Основні компоненти архітектури блокчейн, які забезпечують безпечну передачу текстової інформації між користувачами в мережі:

1. Вузол - користувач або комп'ютер в архітектурі блокчейн (у кожного є незалежна копія всієї книги блокчейн.
2. Трансакція - найменший будівельний блок блокчейн-системи (записи, інформація тощо), який служить метою блокчейну.
3. Блок - структура даних, що використовується для збереження набору транзакцій, яка розподіляється на всі вузли мережі.
4. Ланцюг - послідовність блоків у певному порядку.
5. Шахтарі - конкретні вузли, які виконують процес перевірки блоку, перш ніж додавати що-небудь до структури блокчейна.
6. Консенсус (протокол консенсусу) - сукупність правил та механізмів здійснення блокчейн-операцій.

Будь-який новий запис або транзакція в рамках блокчейна передбачає створення нового блоку. Кожен запис потім перевіряється та підписується цифровим шляхом, щоб забезпечити його справжність. Перш ніж цей блок буде доданий до мережі, його слід перевірити більшістю вузлів у системі. Оскільки структура шифрування та захисту даних в фінансовій сфері дуже схожа на захист текстової інформації, необхідно продемонструвати схему взаємозвязків в ланцюгу блокчейн технологій.

Отже, необхідна схема архітектури блокчейн, яка показує, як це насправді працює у вигляді цифрового гаманця:

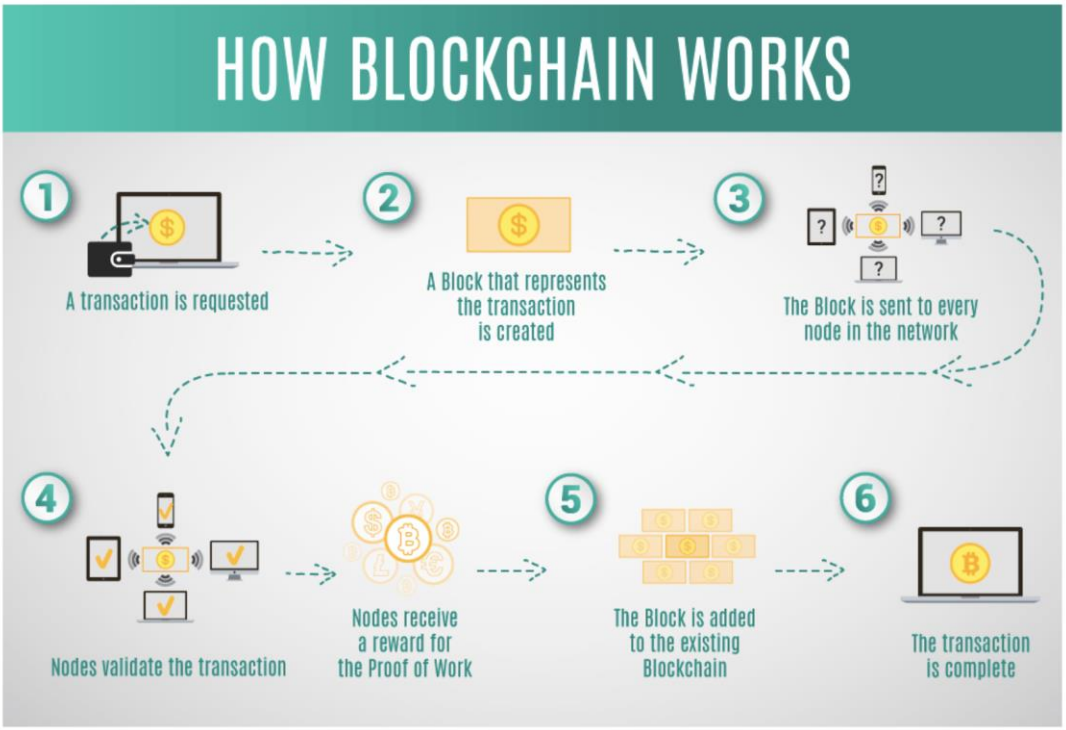


Рис. 3.6. - Схема роботи цифрового гаманця на технології блокчейн

Далі, потрібно детальніше розглянути, що таке блок у блокчейні. Кожен блокчейн-блок складається з:

- певні дані
- хеш блоку
- хеш з попереднього блоку

Дані, що зберігаються всередині кожного блоку, залежать від типу блокчейна. Наприклад, у структурі блокчейна блок підтримує дані про приймач, відправника та кількість монет.

Хеш - це як відбиток пальців (довгий запис, що складається з деяких цифр і літер. Кожен блок хешу генерується за допомогою криптографічного алгоритму хешу (SHA 256). Отже, це допомагає легко ідентифікувати кожен блок у структурі блокчейна. У момент створення блоку він автоматично прикріплює хеш, тоді як

будь-які зміни, внесені в блок, також впливають на зміну хеша. Простіше кажучи, хеши допомагають виявити будь-які зміни блоків:

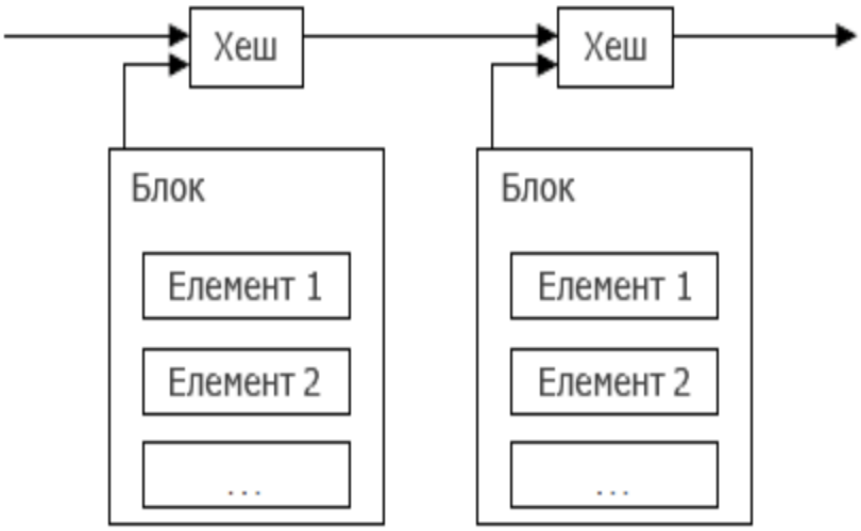


Рис. 3.7. - Схема формування хешу

Кінцевим елементом всередині блоку є хеш з попереднього блоку. Це створює ланцюг блоків і є головним елементом безпеки архітектури блокчейн. Як приклад, заблокуйте 45 балів до блока 46. Перший блок у ланцюжку трохи особливий - усі підтверджені та затверджені блоки отримані з блоку генезису.

Будь-які корупційні спроби провокують зміни блоків. Усі наступні блоки потім несуть невірну інформацію і роблять всю систему блокчейн недійсною.

З іншого боку, теоретично можна було б налаштувати всі блоки за допомогою сильних комп'ютерних процесорів. Однак є рішення, яке виключає цю можливість, яка називається доказом роботи.

Це дозволяє користувачеві уповільнити процес створення нових блоків. В архітектурі блокчейн потрібно близько 10 хвилин, щоб визначити необхідний доказ роботи та додати новий блок до ланцюга. Цю роботу виконують шахтарі - спеціальні вузли в структурі блокчейна Bitcoin. Шахтарі отримують, щоб зберегти плату за транзакції з блоку, який вони підтвердили як винагороду.

Кожен новий користувач (вузол), що приєднується до однорангової мережі блокчейн, отримує повну копію системи. Після створення нового блоку він надсилається до кожного вузла в системі блокчейн.

Потім кожен вузол перевіряє блок і перевіряє правильність зазначеної там інформації. Якщо все гаразд, блок додається до локальної блокчейна в кожному вузлі.

Усі вузли всередині архітектури блокчейн створюють протокол консенсусу. Система консенсусу - це сукупність мережевих правил, і якщо всі дотримуються їх, вони стають самозакоханими всередині блокчейна.

Наприклад, блокчейн Bitcoin має правило консенсусу, яке стверджує, що сума транзакції повинна бути скорочена навпіл після кожні 200 000 блоків. Це означає, що якщо блок виробляє винагороду за перевірку в розмірі 10 BTC, це значення повинно бути вдвічі зменшено через кожні 200 000 блоків.

Крім того, може бути видобуто лише 4 мільйони BTC, оскільки в системі Bitcoin блокчейн протокол закладений максимум 21 мільйон BTC. Після того, як шахтарі розблокують стільки, подача біткойнів закінчується, якщо протокол не буде змінено.

Для резюме це робить технологію блокчейн незмінною та криптографічно захищених, усуваючи будь-які треті сторони. Неможливо підробляти систему blockchain; як потрібно було б підробляти всі його блоки, перерахувати доказ роботи для кожного блоку, а також контролювати понад 50% усіх вузлів у одноранговій мережі.

3.3 Структура передачі даних в блокчейн для шифрування текстової інформації

Структура даних блокчейну - це зв’язаний назад список блоків транзакцій, який упорядковується. Він може зберігатися як плоский файл або у простій базі даних. Кожен блок можна ідентифікувати за допомогою хешу, сформованого за

допомогою алгоритму криптографічного хешування SHA256 у заголовку блоку для побулови даних в блокчейн:

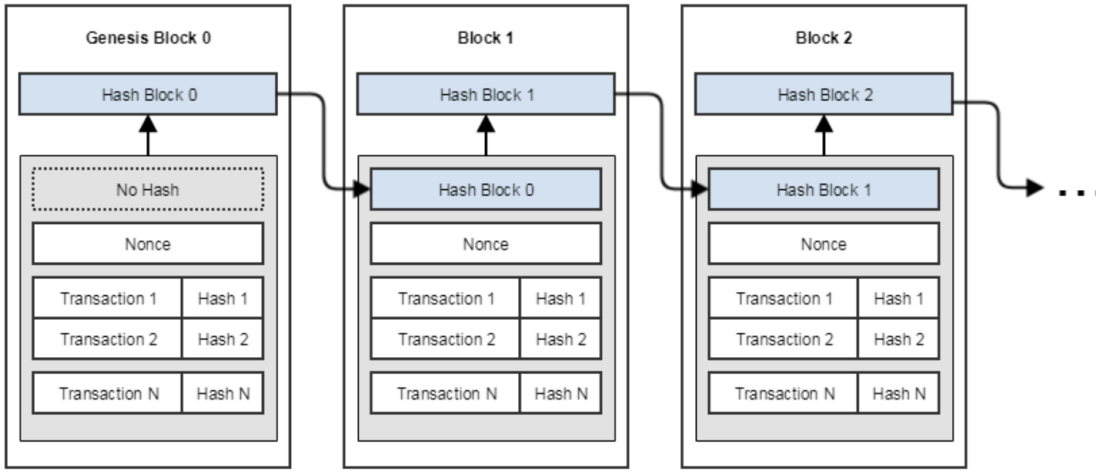


Рис. 3.8. - Схема побудови структури даних блокчейну

Кожен блок посилається на попередній блок, також відомий як батьківський блок, у полі “хеш попереднього блоку” в заголовок блоку.

Хеш, також відомий у довгому вигляді як криптографічна хеш-функція, - це математичний алгоритм, який відображає дані довільного розміру в бітовий рядок фіксованого розміру. У випадку SHA 256 результатом є рядок із 32 байт.

Отримані 32 байти фактично унеможливлюють зворотний вихід, оскільки функція була розроблена як одностороння функція (Schneier, 2004).

Ідея використання хеш-функцій полягає в тому, щоб полегшити ретельні засоби пошуку даних у наборі даних. Найбільш основною формою хеш-функції є будь-яка функція, яка може бути використана для зіставлення даних довільного розміру з даними фіксованого розміру.

Цей результат є бітовим рядком, відомим як хеш-значення, хеш-сума або хеш-код. Хеш-значення можуть зберігатися у табличній формі, відомій як хеш-таблиця, і є ефективним механізмом індексації; особливо корисний у результатах пошуку (Peters & Panayi, 2015).

Хеш-функції теж не мають зіткнень. Це означає, що неможливо знайти два повідомлення, які мають хеш, що має однакову хеш-величину (Narayanan, 2016).

Тому, отримавши компактний хеш, можна підтвердити, що він відповідає певній вихідній точці. Блоки можна ідентифікувати з їх хешу, слугуючи двом цілям; ідентифікація та перевірка цілісності.

Функція хешування біткойнів використовує SHA 256, застосовуваний двічі, див. (Національний інститут стандартів і технологій, 2015). Він генерує майже унікальний 256-бітний (32-байтовий) хеш-захист фіксованого розміру. Великі класи хеш-функцій базуються на будівельному блоці функції стиснення (Peters & Panayi, 2015). Кожен блок містить хеш свого батька всередині власного заголовка.

Там прокладається ланцюжок, що йде назад до першого створеного блоку, також відомого як блок генезису, зв'язаний між собою послідовністю хешів.

Поле "хеш попереднього блоку" знаходиться всередині заголовка блоку, і, отже, поточний хеш блоку залежить від хеш батьківського блоку. Ідентичність дитини змінюється, якщо ідентичність батьків змінюється. Коли будь-який спосіб батьківського модифікується, хеш батьківського елемента змінюється.

Змінений хеш батьків вимагає зміни в покажчику “хеш попереднього блоку” дочірньої організації. Це, в свою чергу, викликає мутацію хешу дитини, що вимагає зміни покажчика онука, що, у свою чергу, змінює онука тощо.

Цей каскадний ефект гарантує, що як тільки блок має багато наступних поколінь, він не може бути змінений без подальшого примусового перерахунку всіх наступних блоків.

Оскільки такий перерахунок вимагав би величезної кількості обчислень, існування довгого ланцюга блоків укріплює глибоку історію Блокчейна, щоб вона була незмінною; ключова особливість безпеки технології блокчейн.

3.4 Побудова блоків в блокчейн

Блок - це структура даних контейнера, яка об'єднує транзакції для включення до загальнодоступної книги, відомої як блокчейн. Блок складається із заголовка; містять метадані, а потім довгий список транзакцій. Блок можна ідентифікувати

двома способами, або шляхом посилання на хеш блоку, або шляхом посилання на висоту блоку.

Заголовок блоку складається з трьох наборів метаданих блоку. Метадані - це дані, що надають інформацію про інші дані.

По-перше, є посилання на хеш попереднього блоку, який з'єднує цей блок з попереднім блоком, що лежить у блокчейні.

Другий набір метаданих стосується гірничої конкуренції; а саме складність, відмітка часу та відсутність.

Нарешті, третьою частиною метаданих є корінь дерева Меркла; структура даних, що використовується для ефективного підсумовування всіх транзакцій у блоці.

Заголовки блоків можна розглядати як приклад багатопартійного підпису з динамічним членством (DMSS). DMSS - це цифровий підпис, утворений набором підписувачів, який не має фіксованого розміру (Back, Corallo, Dashjr, & Friedenbach, 2014).

Заголовки блоків Bitcoin - це DMSS, оскільки їх підтвердження роботи має властивість, яку кожен може внести, не проходячи процес зарахування.

Крім того, внесок зважується пропорційною обчислювальною потужністю, а не одним пороговим внеском підпису на партію (Back, Corallo, Dashjr, & Friedenbach, 2014).

Це дозволяє анонімне членство без ризику атаки Sybil. Напад Sybil - це коли одна сторона приєднується багато разів і має нерівномірний, непропорційний внесок у підпис.

Оскільки блоки зв’язані між собою, DMSS Біткойна є сукупним. Ланцюжок заголовків блоків також є DMSS на своєму першому блоці, з обчислювальною силою, еквівалентною сумі обчислювальних потужностей складаючого DMSS (Back, Corallo, Dashjr, & Friedenbach, 2014). Тому ключовим нововведенням у Blockchain є підпис обчислювальної потужності, а не типовий підпис знань.

3.5 Побудова блока заголовку та вузлів для шифрування інформації задопомогою блокчейн технологій

Наприклад, хеш блоку першого коли-небудь створеного блоку блокчейн - 000000012d7719i085oe235831e934gf763ea46a4c6c191b3f1060a8ce26ac

Хеш блоку ідентифікує блок однозначно і може бути незалежно отриманий будь-яким вузлом просто шляхом хешування заголовка блоку.

Вузол - це повноцінний клієнт. Повноцінний клієнт - це клієнт, який володіє ланцюжками блоків і спільно використовує блоки та транзакції через мережу блокчейнів. Вузол вважається частиною інфраструктури блокчейну і не обов'язково повинен бути майнером.

Кожен вузол зберігає повну копію повністю упорядкованої послідовності подій у вигляді блокчейну.

Хеш блоку обчислюється кожним вузлом, оскільки блок отримується від мережі. Хеш блоку може зберігатися в окремій таблиці бази даних як частина метаданих блоку, щоб полегшити індексацію та швидший пошук блоків з диска.

Висота блоку - це ще один метод ідентифікації блоку, на цей раз через його позицію в блокчейні. Перший коли-небудь створений блок знаходиться на висоті блоку 0 (нуль), а у випадку з біткойнами це той самий блок, на який посилався хеш блоку 000000012d7719i085oe235831e934gf763ea46a4c6c191b3f1060a8ce26ac вище.

Кожен наступний блок, доданий "зверху" цього першого блоку, знаходиться на одній позиції "вище" у блокчейні, подібно до ящиків, складених один над одним. Висота блоку не завжди ідентифікує конкретний одиничний блок. Можливо, два або більше блоків можуть мати однакову висоту блоку, обидва конкуруючи за однакову позицію в блокчейні.Перший блок у будь-якому блокчейні називається блоком генезису. Якщо ви почнете з будь-якого блоку і хронологічно стежите за ланцюжком назад, ви дійдете до блоку генезису. Блок генезису статично кодується в клієнтському програмному забезпеченні, тому його неможливо змінити. Кожен вузол може ідентифікувати хеш і структуру блоку генезису, фіксований час

створення та окремі транзакції всередині. Таким чином, кожен вузол має безпечний “корінь”, з якого можна побудувати надійний блокчейн.

3.6 Пов’язування блоків у блокчейні

Вузли зберігають копію блокчейну локально, починаючи з блоку генезису. Локальна копія блокчейну постійно оновлюється, коли нові блоки виявляються і згодом будуються на ланцюжку.

Оскільки вузол отримує інформацію про вхідні блоки з мережі, він спочатку перевірить ці блоки, а потім зв’яже їх із існуючим блокчейном. Процес встановлення зв'язку такий: вузол перевірить заголовок вхідного блоку та шукає “хеш попереднього блоку”. Переглядаючи цей вхідний блок, вузол знаходить поле «хеш попереднього блоку», що містить хеш свого батьківського блоку. Цей хеш відомий вузлу раніше.

Отже, вузол вважає, що цей новий блок є нащадком останнього блоку в ланцюжку, і є законним продовженням ланцюжка. Вузол додає цей новий блок до кінця ланцюжка, роблячи блокчейн довшим з новою висотою вхідного блоку, який зараз перевіряється. Кожен блок у блокчейні містить підсумок усіх транзакцій у блоці, використовуючи дерево Merkle. Дерево Меркла - це структура даних, яка використовується для ефективного узагальнення та перевірки цілісності великих наборів даних.

Дерева меркле також відомі як двійкове хеш-дерево. Дерева Меркле - це бінарні дерева, що містять криптографічні хеші. Термін "дерево" походить від області інформатики, що описує розгалужену структуру даних. Дерева Меркле видають загальний цифровий відбиток пальців усього набору транзакцій. Дерево Меркле будується шляхом рекурсивного хешування пар вузлів, поки не буде лише одного хешу, який називається корінь або корінь Меркле. Криптографічний хеш-алгоритм, що використовується у символічних біткойнових деревах Меркла, - це SHA256, застосовуваний двічі, також відомий як double-SHA256. SHA - це алгоритм безпечного хешування, який є захищеним набором криптографічних хеш-

функцій, у сімействі SHA-2. Коли N елементів елементів хешуються та узагальнюються в Дереві Меркла, ви можете перевірити, чи включений певний елемент до дерева із щонайбільше 2 * log2 (N) кількістю обчислень, що забезпечує дуже ефективний спосіб перевірити, чи транзакція дійсно включається в блок.

Дерево Меркле побудовано знизу вгору. Ми починаємо з чотирьох транзакцій; позначаються як Tx A, Tx B, Tx C, Tx D. Ці транзакції не зберігаються в дереві Меркла, швидше їх дані хешуються, а отриманий хеш зберігається в кожному листовому вузлі як HA, HB, HC та HD.

Математичну функцію для виведення HA можна розглядати як HA = SHA256 (SHA256 (транзакція A)), де транзакція A була криптографічно хешована двічі за допомогою SHA256. Потім послідовні пари вузлів об'єднуються в батьківський вузол, об'єднуючи два хеші та хешуючи їх разом. За прикладом, щоб побудувати батьківський вузол HAB, два 32-байтових хеші дочірніх об'єднуються, щоб створити 64-байтовий рядок. Потім цей рядок подвійно хешується, щоб отримати хеш батьківського вузла:

HAB = SHA 256 (SHA 256 (HA + HB))

Дерево меркле - це бінарне дерево. У випадку, якщо для узагальнення існує непарна кількість транзакцій, хеш останньої транзакції буде продубльовано. Процес триває вгору, доки вгорі не буде лише одного вузла. Цей вузол відомий як корінь Меркла. 32-байтовий хеш зберігається в заголовку блоку і узагальнює всі дані за всіма чотирма транзакціями. За допомогою цього методу дерево Merkle може підсумовувати будь-яку кількість транзакцій у блоці до 32 байт.

3.7 Доведення транзакцій у блоках

При спробах з'ясувати, чи включена певна транзакція в блок, можна легко виявити шлях автентифікації або шлях Merkle, який з'єднує конкретну транзакцію з коренем дерева. Вузол математично вимагався б лише для створення 32-байтових хешів log2 (N). Це критично важливо, оскільки логарифм числа 2 для бази збільшується набагато повільніше, ніж просто число. Отже, це означає, що вузлу

потрібно лише створити шляхи по 10 або 12 хешів (від 320 до 384 байт), щоб довести одну транзакцію з понад тисячі транзакцій у блоці.

Контракти, транзакції, і записи їх серед визначальних структур в наших економічних, законних, і політичних системах. Вони захищають активи і встановлюють організаційні межі. Вони встановлюють і перевіряють тотожність і події хроніки.

Вони регулюють взаємодії серед націй, організацій, співтовариств, і індивідуумів. Вони ведуть директорську і соціальну дію. І все ж ці критичні інструментарії і бюрократія сформували, щоб управляти ними не тримали з цифровою трансформацією економіки.

Вони подібні до затору години натиску, організація налагоджувальних переривань Формули 1, мчать маршрутизацію з централізованим доступом. У цифровому світі, шлях, який ми регулюємо і підтримуємо адміністративний елемент управління, доводиться змінитися. Програма фінансового обліку безпосередньо може також бути програмованою, щоб запустити транзакції автоматично. За допомогою блокчейну ми можемо уявити світ, в якому контракти вбудовуються в цифровий код і зберігаються у прозорих спільних базах даних, де вони захищені від видалення, фальсифікації та перегляду. У цьому світі кожна угода, кожен процес, кожне завдання та кожен платіж мали б цифровий запис та підпис, які можна було б ідентифікувати, перевірити, зберегти та надати спільний доступ. Посередники, такі як юристи, брокери та банкіри, можуть більше не знадобитися. Люди, організації, машини та алгоритми могли б вільно взаємодіяти та взаємодіяти між собою з невеликим тертям. Це величезний потенціал блокчейну.

Дійсно, практично всі чули твердження про те, що блокчейн зробить революцію в бізнесі та перегляне компанії та економіку. Незважаючи на те, що ми поділяємо ентузіазм щодо його потенціалу, нас турбує ажіотаж. Нас турбують не лише проблеми безпеки (наприклад, крах однієї біржової біржі в 2014 році та нові хакерські атаки інших).

Було б помилкою кидатися з головою в блокчейн-інновації, не розуміючи, як це, швидше за все, закріпиться.

Ми впевнені, що до справжньої трансформації бізнесу та уряду, що керується блокчейном, ще багато років.

Це пов’язано з тим, що блокчейн - це не “руйнівна” технологія, яка може атакувати традиційну бізнес-модель за допомогою більш дешевого рішення та швидко наздоганяти діючі фірми.

Blockchain - це фундаментальна технологія: вона може створити нові основи для наших економічних та соціальних систем. Але хоча вплив буде величезним, блокчейн займе десятки років, щоб проникнути в нашу економічну та соціальну інфраструктуру.

Процес впровадження буде поступовим і стабільним, а не раптовим, оскільки хвилі технологічних та інституційних змін набирають обертів. Одним з найбільш актуальних прикладів є технологія розподілених комп’ютерних мереж, що спостерігається у прийнятті протоколу TCP / IP (протокол управління передачею / Інтернет-протокол), який заклав основу для розвитку Інтернету. Запроваджений у 1972 році, TCP / IP вперше отримав популярність у випадку одноразового використання: як основа для електронної пошти серед дослідників на ARPAnet, попереднику комерційного Інтернету Міністерства оборони США. До TCP / IP телекомунікаційна архітектура базувалася на "комутації каналів", в якій зв'язки між двома сторонами або машинами повинні були бути попередньо встановлені та підтримуватися протягом обміну.

Щоб забезпечити можливість спілкування будь-яких двох вузлів, постачальники телекомунікаційних послуг та виробники обладнання інвестували мільярди у створення спеціальних ліній. TCP / IP перевернув цю модель. Новий протокол передавав інформацію, оцифровуючи її та розбиваючи на дуже маленькі пакети, кожен з яких включав інформацію про адресу. Після випуску в мережу пакети можуть пройти будь-який шлях до одержувача. Розумні вузли надсилання та прийому на краях мережі можуть розібрати та повторно зібрати пакети та інтерпретувати закодовані дані. Не було потреби у виділених приватних лініях або масивній інфраструктурі. TCP / IP створив відкриту спільну загальнодоступну мережу без будь-якого центрального органу чи сторони, відповідального за її

обслуговування та вдосконалення. Традиційні телекомунікаційні та обчислювальні сектори дивилися на TCP / IP скептично. Мало хто уявляв, що на новій архітектурі можуть бути встановлені надійні з'єднання даних, обміну повідомленнями, голосовими та відеозв'язками або що пов'язана з цим система може бути захищена і масштабована. Але наприкінці 1980-х та 1990-х років все більше фірм, таких як Sun, NeXT, Hewlett-Packard та Silicon Graphics, використовували TCP / IP, частково для створення локалізованих приватних мереж в організаціях.

Для цього вони розробили будівельні блоки та інструменти, які розширили його використання за межі електронної пошти, поступово замінюючи більш традиційні технології та стандарти локальної мережі. Коли організації приймали ці будівельні блоки та інструменти, вони бачили значний приріст продуктивності.

TCP / IP отримав широке загальнодоступне використання з появою Всесвітньої павутини в середині 90-х. Швидко з’явилися компанії нових технологій, які надали «сантехніку» - апаратне забезпечення, програмне забезпечення та послуги, необхідні для підключення до загальнодоступної мережі та обміну інформацією.

Netscape комерціалізував браузери, веб-сервери та інші інструменти та компоненти, які сприяли розробці та впровадженню Інтернет-сервісів та додатків. Sun стимулював розробку Java, мови програмування додатків. Оскільки інформація в Інтернеті зростала в геометричній прогресії, Infoseek, Excite, AltaVista та Yahoo народились для керівництва користувачами навколо неї.

Як тільки ця базова інфраструктура набрала критичної маси, нове покоління компаній скористалося недорогими зв’язками, створивши Інтернет-послуги, які стали привабливими замінниками існуючого бізнесу. CNET перемістив новини в Інтернет. Amazon запропонував на продаж більше книг, ніж будь-який книжковий магазин.

Priceline та Expedia спростили купівлю авіаквитків та внесли безпрецедентну прозорість процесу. Здатність цих новачків отримати широкий охоплення за відносно низькою вартістю зробила значний тиск на такі традиційні компанії, як газети та роздрібні магазини.

Спираючись на широке підключення до Інтернету, наступна хвиля компаній створила нові, трансформаційні програми, які принципово змінили спосіб створення бізнесу та отримання цінності.

Ці компанії були побудовані на новій одноранговій архітектурі та генерували цінність шляхом координації розподілених мереж користувачів. Подумайте, як eBay змінив роздрібну торгівлю через аукціони, Napster - музичну індустрію, Skype - телекомунікації, а Google, яка використовувала створені користувачами посилання для отримання більш відповідних результатів, - веб-пошук.

Зрештою, потрібно було більше 30 років, щоб TCP / IP пройшов усі етапи - одноразове використання, локалізоване використання, заміщення та перетворення - та змінив економіку.

Сьогодні більше половини найцінніших державних компаній у світі мають бізнес-моделі, що базуються на Інтернеті. Змінилися самі основи нашої економіки.

Фізичний масштаб та унікальна інтелектуальна власність більше не надають неперевершених переваг; дедалі більше економічними лідерами є підприємства, які виступають як “ключові камені”, активно організовуючи, впливаючи та координуючі розповсюджені мережі громад, користувачів та організацій. Blockchain - однорангова мережа, яка знаходиться на вершині Інтернету - була введена в жовтні 2008 року як частина пропозиції щодо біткойнів, системи віртуальної валюти, яка ухилялася від центрального органу з питань випуску валюти, передачі права власності та підтвердження транзакцій. Біткойн - це перше застосування технології блокчейн.

Паралелі між блокчейном та TCP / IP очевидні. Подібно до електронної пошти, що підтримує двосторонні повідомлення, біткойн дозволяє двосторонні фінансові операції. Розробка та підтримка блокчейну є відкритою, розподіленою та спільною - подібно до TCP / IP. Команда добровольців по всьому світу підтримує основне програмне забезпечення. І так само, як електронна пошта, біткойни спочатку впізнали захоплену, але відносно невелику спільноту. TCP / IP відкрив нову економічну цінність, різко знизивши вартість з'єднань. Подібним чином

блокчейн може різко знизити вартість транзакцій. Він може стати системою записів для всіх транзакцій.

Якщо це станеться, економіка знову зазнає кардинальних зрушень, оскільки з’являються нові джерела впливу та контролю, засновані на блокчейні. Ведення постійних записів про операції є основною функцією будь-якого бізнесу. Ці записи відстежують минулі дії та результати діяльності та керують плануванням на майбутнє. Вони дають уявлення не лише про те, як організація працює внутрішньо, а й про зовнішні стосунки організації. Кожна організація веде власні записи, і вони є приватними. Багато організацій не мають головної книги всієї своєї діяльності; натомість записи розподіляються між внутрішніми блоками та функціями. Проблема полягає в тому, що узгодження операцій між окремими та приватними книгами займає багато часу і схильне до помилок.

Наприклад, типова біржова операція може бути виконана протягом мікросекунд, часто без втручання людини. Однак врегулювання - передача права власності на акції - може зайняти до тижня. Це пов’язано з тим, що сторони не мають доступу до бухгалтерських книг одна одної і не можуть автоматично підтвердити, що активи фактично належать і можуть бути передані.

Натомість низка посередників виступають гарантами активів, оскільки запис транзакцій, що обертаються організаціями, та книги реєстру оновлюються індивідуально.

У системі блокчейнів книга реєструється у великій кількості однакових баз даних, кожна з яких розміщується та підтримується зацікавленою стороною. Коли зміни вводяться в одну копію, усі інші копії одночасно оновлюються.

Отож, коли відбуваються операції, записи про обмін вартості та активів постійно вносяться у всі книги. Не потрібно стороннім посередникам перевіряти або передавати право власності.

Якби біржова транзакція відбулася в системі, що базується на блокчейні, вона була б врегульована протягом декількох секунд, надійно та достовірно. (Ганебні хаки, які потрапили на біткойн-біржі, виявили слабкі місця не в самому блокчейні, а в окремих системах, пов'язаних із сторонами, що використовують блокчейн.)

Якщо біткойн схожий на ранню електронну пошту, чи може блокчейн десятиліття досягти свого повного потенціалу. Ми не можемо точно передбачити, скільки років займе трансформація, але ми можемо здогадуватися, які типи додатків набудуть першої популярності і як з часом стане загальновизнаним прийняття блокчейну, які поділяються на два типи.

Перший - це новизна - ступінь, в якому заявка є новою для світу. Чим новіший він, тим більше зусиль буде потрібно для того, щоб користувачі зрозуміли, які проблеми він вирішує.

Другий - це складність, представлена рівнем залученої координації екосистем - кількістю та різноманітністю сторін, яким потрібно співпрацювати, щоб отримати цінність із технологією. Наприклад, соціальна мережа з одним учасником мало корисна; соціальна мережа варта лише тоді, коли до неї підписано багато ваших власних зв'язків. Інші користувачі програми повинні бути залучені на борт, щоб створити цінність для всіх учасників. Те саме стосуватиметься багатьох програм блокчейну. І, оскільки масштаби та вплив цих заявок збільшуються, їх прийняття потребуватиме значних інституційних змін.

Кожен квадрант являє собою етап розвитку технологій. Визначення того, в який саме блокчейн-інновація потрапляє, допоможе керівникам зрозуміти типи викликів, які воно представляє, рівень співпраці та консенсусу, які йому потрібні, а також законодавчі та регуляторні зусилля, які йому знадобляться.

Карта також пропонує, які процеси та інфраструктура повинні бути створені для сприяння впровадженню інновації. Менеджери можуть використовувати його для оцінки стану розвитку блокчейну в будь-якій галузі, а також для оцінки стратегічних інвестицій у власні можливості блокчейну.

У першому квадранті розташовані програми з низькою новизною та низькою координацією, які створюють кращі, менш затратні та цілеспрямовані рішення. Електронна пошта, дешева альтернатива телефонним дзвінкам, факсам та поштовій пошті, була одноразовою програмою для TCP / IP (хоча її значення зростало із збільшенням кількості користувачів).

Біткойн теж потрапляє в цей квадрант. Навіть у перші дні біткойн пропонував негайну цінність тим небагатьом людям, які використовували його просто як альтернативний спосіб оплати. (Ви можете сприймати це як складний електронний лист, який передає не лише інформацію, а й фактичну вартість.)

Це все ще помилка округлення порівняно з 411 трильйонами доларів загального обсягу глобальних платежів, але біткойн зростає швидко і стає все більш важливим у таких контекстах, як миттєві платежі та торгівля іноземною валютою та активами, де нинішня фінансова система має обмеження.

3.8 Локалізація блокчейн для шифрування тектової інформації

Другий квадрант включає інновації, які мають відносно високу новизну, але для створення негайної цінності потребують лише обмеженої кількості користувачів, тому сприяти їх прийняттю все ще досить просто.

Якщо блокчейн слідує шляху мережевих технологій, прийнятих у бізнесі, ми можемо очікувати, що інновації блокчейну будуть побудовані на одноразових додатках для створення локальних приватних мереж, в яких кілька організацій пов’язані через розподілену книгу:

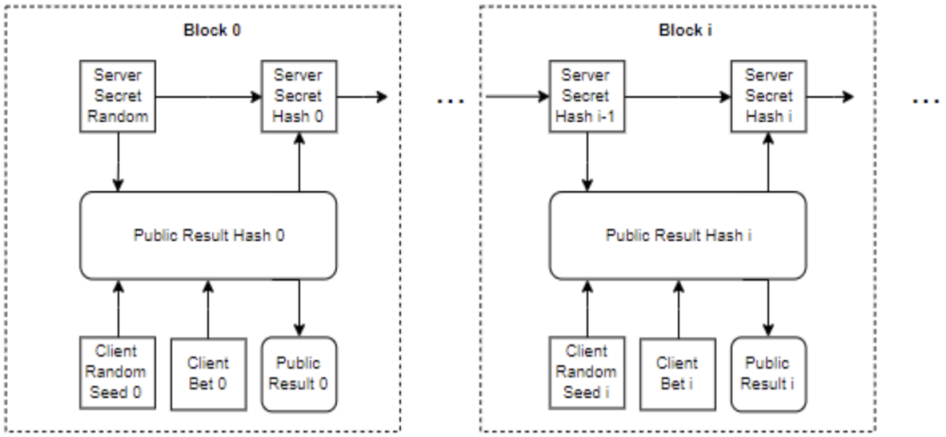


Рис. 3.9. - Схема локалізації блокчейн

Більша частина початкового розвитку приватного блокчейну відбувається у секторі фінансових послуг, часто в невеликих мережах фірм, тому вимоги до

координації відносно скромні. Nasdaq співпрацює з Chain.com, одним з багатьох постачальників інфраструктури блокчейнів, пропонуючи технологію обробки та перевірки фінансових операцій.

Банк Америки, JPMorgan, Нью-Йоркська фондова біржа, Fidelity Investments і Standard Chartered тестують технологію блокчейну як заміну паперовій та ручній обробці транзакцій у таких сферах, як торговельне фінансування, іноземна валюта, транскордонне врегулювання та цінні папери поселення. Банк Канади тестує цифрову валюту під назвою CAD-coin для міжбанківських переказів. Ми передбачаємо розповсюдження приватних блокчейнів, які слугують певним цілям для різних галузей.

Третій квадрант містить програми, які мають порівняно низьку новизну, оскільки вони базуються на існуючих одноразових та локалізованих додатках, але мають високі потреби у координації, оскільки вони залучають до більш широкого та все більш публічного використання.

Ці інновації мають на меті замінити цілі способи ведення бізнесу. Однак вони стикаються з високими бар'єрами для усиновлення; їм не тільки потрібна більша координація, але процеси, які вони сподіваються замінити, можуть бути повномасштабними та глибоко вбудованими в організації та установи.

Прикладами замінників є криптовалюти - нові, повністю сформовані валютні системи, які виросли із простої технології оплати біткойнами. Критична відмінність полягає в тому, що криптовалюта вимагає від кожної сторони, яка здійснює грошові операції, прийняти її, кидаючи виклик урядам та установам, які давно обробляли та контролювали такі операції. Споживачі також повинні змінити свою поведінку та зрозуміти, як впровадити нові функціональні можливості криптовалюти.

Нещодавній експеримент на MIT висвітлює виклики, що стоять перед системами цифрових валют. У 2014 році біткойн-клуб MIT забезпечив кожного з 4 494 студентів MIT 100 доларів у біткойнах. Цікаво, що 30% студентів навіть не підписалися на безкоштовні гроші, а 20% реєстрацій конвертували біткойн в

готівку протягом декількох тижнів. Навіть технічно підкованим було важко зрозуміти, як і де використовувати біткойн.

Одним з найамбітніших додаткових блокчейнових додатків є Stellar, некомерційна організація, яка має на меті забезпечити доступними фінансові послуги, включаючи банку, мікроплатежі та грошові перекази, людям, які ніколи до них не мали доступу. Stellar пропонує власну віртуальну валюту, люмен, а також дозволяє користувачам зберігати в своїй системі цілий ряд активів, включаючи інші валюти, телефонні хвилини та кредити даних.

Спочатку Stellar зосередився на Африці, особливо на Нігерії, найбільшій там економіці. Він бачив значне прийняття серед своєї цільової групи та довів свою економічну ефективність. Але його майбутнє аж ніяк не певне, оскільки проблеми з координацією екосистем високі. Незважаючи на те, що прийняття низового рівня продемонструвало життєздатність Stellar, щоб стати банківським стандартом, йому потрібно буде впливати на державну політику та переконувати центральні банки та великі організації використовувати її. Це може зайняти роки узгоджених зусиль.

В останній квадрант потрапляють абсолютно нові програми, які у разі успіху можуть змінити саму природу економічних, соціальних та політичних систем. Вони передбачають координацію діяльності багатьох суб'єктів та досягнення інституційної згоди щодо стандартів та процесів. Їх прийняття потребуватиме серйозних соціальних, правових та політичних змін.

«Розумні контракти» можуть бути найбільш трансформаторним додатком блокчейну на даний момент. Ці автоматизовані платежі та переказ валюти чи інших активів за умови домовленості. Наприклад, смарт-контракт може надіслати платіж постачальнику відразу після доставки вантажу.

Фірма може сигналізувати через блокчейн про те, що конкретний товар отримано - або продукт може мати функціональність GPS, яка автоматично реєструє оновлення місцезнаходження, яке, в свою чергу, спричинило платіж. Ми вже бачили кілька ранніх експериментів із такими самовиконуючимися контрактами у сферах венчурного фінансування, банківської справи та управління

цифровими правами. Наслідки захоплюючі. Фірми будуються на контрактах, від корпорації до відносин з покупцем та постачальником до відносин з працівниками.

Якщо договори автоматизовані, то що буде з традиційними фірмовими структурами, процесами та посередниками, такими як юристи та бухгалтери? А як щодо менеджерів? Всі їх ролі кардинально змінилися б. Перш ніж ми будемо занадто хвилюватися тут, давайте пам’ятати, що нас чекає десятиліття від широкого прийняття смарт-контрактів. Вони не можуть бути ефективними, наприклад, без інституційного бай-ін. Потрібна надзвичайна ступінь координації та чіткості того, як розробляються, перевіряються, виконуються та виконуються смарт-контракти. Ми віримо, що інституції, відповідальні за ці грізні завдання, будуть довго розвиватися.

Висновок до розділу 3

В роботі з архітектурою блокчейн технологій було визначено структуру методів захисту текстової інформації, що буде передаватись між користувачами в мережі. Завдяки детальному використанню структури передачі даних в блокчейн для шифрування текстової інформації, побудові блоків в блокчейн, пов’язування блоків у блокчейні, локалізація блокчейн, буде виконано розробку найбільш ефективних методів захисту текстової інформації між користувачами в мережі за допомогою блокчейн технологій.

4 ЗАСТОСУВАННЯ МЕТОДІВ ШИФРУВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ ЗА ДОПОМОГОЮ БЛОКЧЕЙН ТЕХНОЛОГІЙ

4.1 Початок реалізації методів шифрування текстової інформації на базі блокчейн

Основна концепція блокчейна досить проста: розподілена база даних, яка підтримує постійно зростаючий список впорядкованих записів. Проте, многое залишається незрозумілим, коли ми говоримо про блокчейне, так само залишається багато проблем, які ми намагаємося вирішити з його допомогою. Це відноситься і до популярних блокчейн проектів, таким як біткоїн (Bitcoin) і ефіріум (Ethereum).

Термін "блокчейн" зазвичай сильно прив'язаний до концепції типу грошових переказів, смарт-контрактов або криптовалюты. Це робить розуміння блокчейна складніше, ніж є насправді. Для реалізації методів шифрування даних потрібно розробити відповідну програму для безпечної передачі текстової інформації в мережі. Перший логічний крок - визначитися із структурою блоку. Щоб залишити все як можна простіше, ми включили тільки найнеобхідніше: індекс, відмітка, дані, хэш і хэш попереднього блоку:

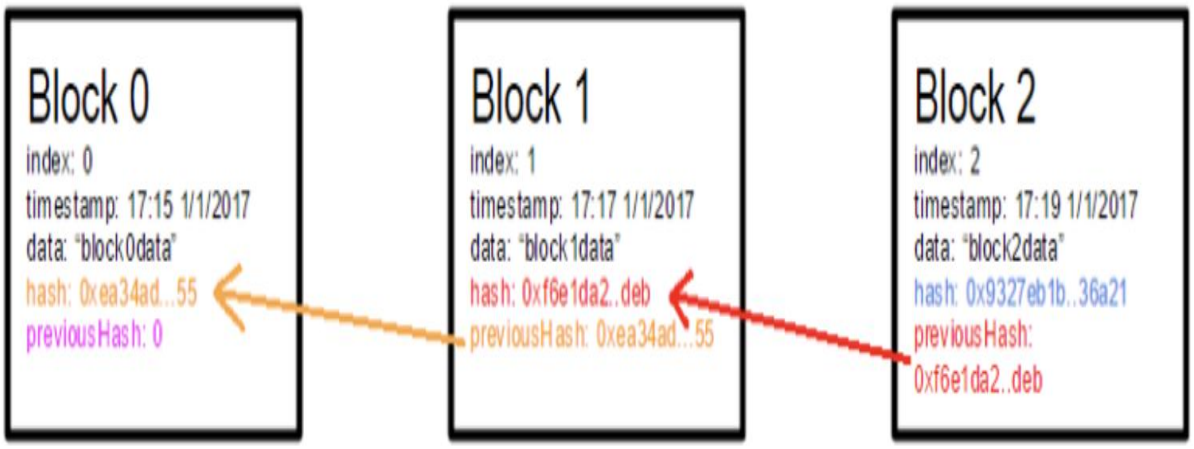


Рис. 4.1. - Структура передачі даних між блоками в блокчейн

Отже, саме за такою схемою потрібно буде реалізувати передачу даних між блокчейн блоками.

4.2 Розробка хеш-блоку для шифрування текстової інформації

Хэш попереднього блоку необхідно знайти у блоці для збереження цілісності ланцюга:

```
class Block {
    constructor(index, previousHash, timestamp, data, hash) {
        this.index = index;
        this.previousHash = previousHash.toString();
        this.timestamp = timestamp;
        this.data = data;
        this.hash = hash.toString();
    }
}
```

Рис. 4.2. – Хеш блоку #1 для шифрування в блокчейн

Блок має бути захешований, щоб зберегти цілісність даних. SHA256 відповідає за зміст блоку. Слід зазначити, що цей хэш не має нічого спільного з "майнингом", оскільки немає підтвердження роботи - рішення задачі:

```
var calculateHash = (index, previousHash, timestamp, data) => {
    return CryptoJS.SHA256(index + previousHash + timestamp + data).toString();
};
```

Рис. 4.3. – Хеш блоку #2 для шифрування в блокчейн

Саме такий блок потрібно додати для зв’язування ланцюгів при шифруванні за допомогою блокчейн технологій.

4.3 Генерація блоку для передачі інформації кінцевому користувачу

Для створення блоку треба знати хэш попереднього блоку, а решту необхідно створювати з наступного змісту (= index, hash, data і timestamp). Такий зміст реалізується для цілісності даних що будуть передаватись в зашифрованих блоках, та такий зміст буде надавати інформацію попереднім блокам про місцезнаходження нових блоків в блокчейн. Таким чином потрібно реалізувати блок-дату. Після реалізації їх потрібно буде звязатти з попередніми блоками в ланцюгах. Для полегшення знаходження блоків, їх потрібно буде ідентифікувати та звязати між собою. Отже, необхідно реалізувати передачу в блок-даті між

користувачами. Блок-дата - це деяка інформація, яка передається кінцевому користувачеві:

```
var generateNextBlock = (blockData) => {
    var previousBlock = getLatestBlock();
    var nextIndex = previousBlock.index + 1;
    var nextTimestamp = new Date().getTime() / 1000;
    var nextHash = calculateHash(nextIndex, previousBlock.hash,
nextTimestamp, blockData);
    return new Block(nextIndex, previousBlock.hash, nextTimestamp,
blockData, nextHash);
};
```

Рис. 4.4. – Створення змісту index, hash, data і timestamp

Отже, після створення блоку, необхідно провести зберігання та перевікру цілісності блоків блокчейн.

4.4 Зберігання та перевірка цілісності блоків блокчейн

В пам'яті масив JavaScript використовується для зберігання блокчейн. Перший блок у блокчейн - це завжди так званий "генезис-блок", що має наступний код:

```
var getGenesisBlock = () => {
    return new Block (0, "0", 1465154705, "my genesis block!!",
"816534932c2b7154836da6afc367695e6337db8a921823784c14378abed4f7d7");
};

var blockchain = [getGenesisBlock ()];
```

Рис. 4.5. – Реалізація генезис-блоку

У будь-який момент часу ми маємо бути в змозі перевірити, чи є блок або ланцюжок блоків допустимим з точки зору цілісності. Отже, після реалізації генезис-блоку потрібно реалізувати отримання інформації з інших блоків та інших

вузлів. Це особливо актуально, коли ми отримуємо нові блоки від інших вузлів і повинні вирішити, приймати їх або ні:

```
var isValidNewBlock = (newBlock, previousBlock) => {
    if (previousBlock.index + 1 !== newBlock.index) {
        console.log('invalid index');
        return false;
    } else if (previousBlock.hash !== newBlock.previousHash) {
        console.log('invalid previoushash');
        return false;
    } else if (calculateHashForBlock(newBlock) !== newBlock.hash) {
        console.log('invalid hash: ' + calculateHashForBlock(newBlock) +
newBlock.hash);
        return false;
    }
    return true;
};
```

Рис. 4.6. – Реалізація нових блоків даних в блокчейн

Завжди має бути тільки один явний набір блоків в ланцюзі водночас часу. У разі виникнення конфліктів (наприклад, два вузли як в створеному блоці № 72) ми вибираємо ланцюг, який має щонайдовший ряд блоків.

Задля коректної роботи системи необхідно зробити коректне формування ланцюгів між блоками, щоб забезепечити надійне шифрування текстової інформації та запобігти супереченням між блоковими структурами в блокчейн, а також, надати сигнал блокам про взаємодію між ними та іншою частиною блокчейну.

Отже, потрібно сформувати схему ланцюгів:

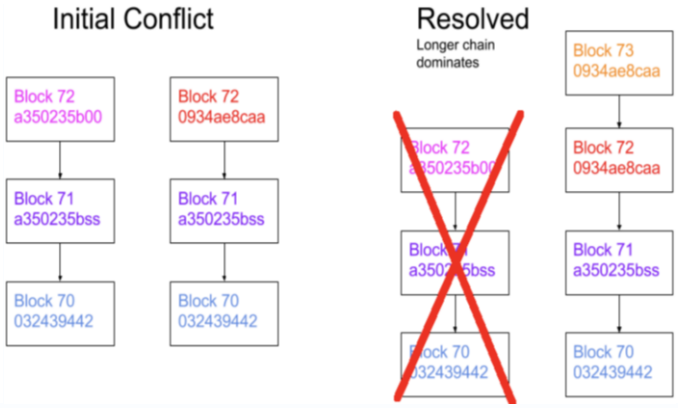


Рис. 4.7. - Формування ланцюгів між блоками в блокчейн

Таким чином, розуміючи алгоритм вірного формування ланцюгів між блоками в блокчейн, можна приступати до надання вузлам необхідного функціоналу.

Наступним кроком буде надання функціоналу вузлам, які будуть об’єднувати необхідні нам блоки в блокчейн:

```
var replaceChain = (newBlocks) => {
    if (isValidChain(newBlocks) && newBlocks.length > blockchain. length) {
        console.log ('Received blockchain is valid. Replacing current blockchain
with received blockchain');
        blockchain = newBlocks;
        broadcast(responseLatestMsg ());
    } else {
        console.log ('Received blockchain invalid');
    }
};
```

Рис. 4.8. - Формування функцій вузла в блокчейн

Важливою функцією вузла є - розділення і синхронізація блокчейн з іншими вузлами. Правила - використовувані для підтримки синхронізації мережі:

- Коли вузол генерує новий блок, він транслює його в мережу
- Коли вузол підключається до нової однорангової мережі він спирається на останній блок
- Коли вузол виявляє блок, який має індекс більший, ніж поточний відомий блок, він або додає блок в його нинішній стані у свій власний ланцюг або підтримує для заповнення блокчейна:

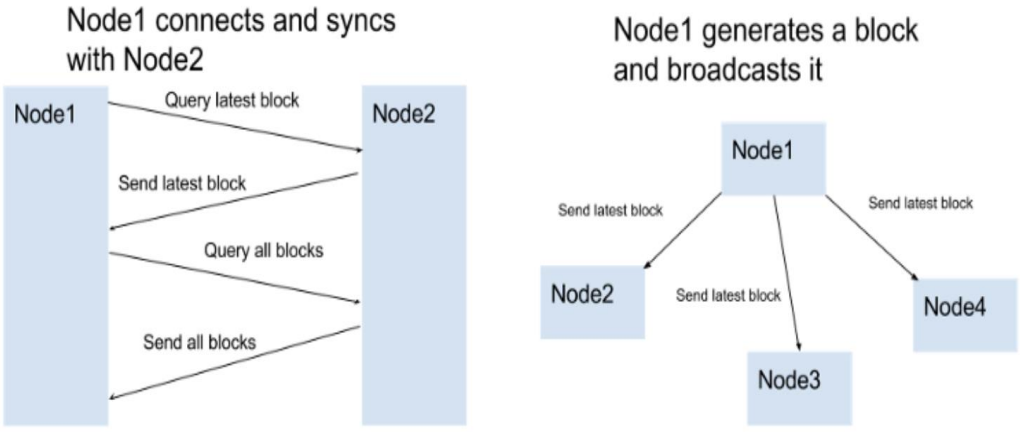


Рис. 4.9. - Зв'язок між блочними елементами

Деякі типові комунікаційні сценарії, які слідують, коли вузли підкорятися описаному протоколу. Немає ніякого автоматичного взаємного виявлення. Місця

(URL - адреси) розташування сторони мають бути додані вручну. Користувач, в деякому роді, повинен мати можливість контролювати вузол. Це робиться шляхом налаштування http-сервера:

```
var initHttpServer = () => {
    var app = express ();
    app.use (bodyParser.json());

    app.get ('/blocks', (req, res) => res. send (JSON.stringify(blockchain)));
    app.post ('/mineBlock', (req, res) => {
        var newBlock = generateNextBlock (req. body.data);
        addBlock(newBlock);
        broadcast (responseLatestMsg ());
        console.log ('block added: ' + JSON.stringify(newBlock));
        res. send ();
    });
    app.get ('/peers', (req, res) => {
        res.   send(sockets.map(s   =>   s._socket.remoteAddress   +   ':'   +
s._socket.remotePort));
    });
    app.post ('/addPeer', (req, res) => {
        connectToPeers ([req. body.peer]);
        res. send();
    });
    app.  listen(http_port,  ()  =>  console.log('Listening  http  on  port:  '  +
http_port));
};
```

Рис. 4.10. – Налаштування зв’язку з http-сервером

Як видно, користувач може взаємодіяти з вузлом наступними способами:

1. Переглядати список усіх блоків
2. Створювати новий блок зі змістом, заданим користувачем
3. Переглядати або додавати однорангових користувачів
4. Найбільш простий спосіб управління вузлом

4.5 Реалізація логіну до обміну текстовою інформацією задопомогою блокчейн

Слід зазначити, що вузол фактично надає два веб-сервери: один для користувача, щоб контролювати вузол(http-сервер), для повноцінного функціонування в мережі.

А також, один для однорангового (peer - to - peer) зв'язку між вузлами (websocket сервер http).

Тож потрібно реалізувати схему мережевої передачі блоків:

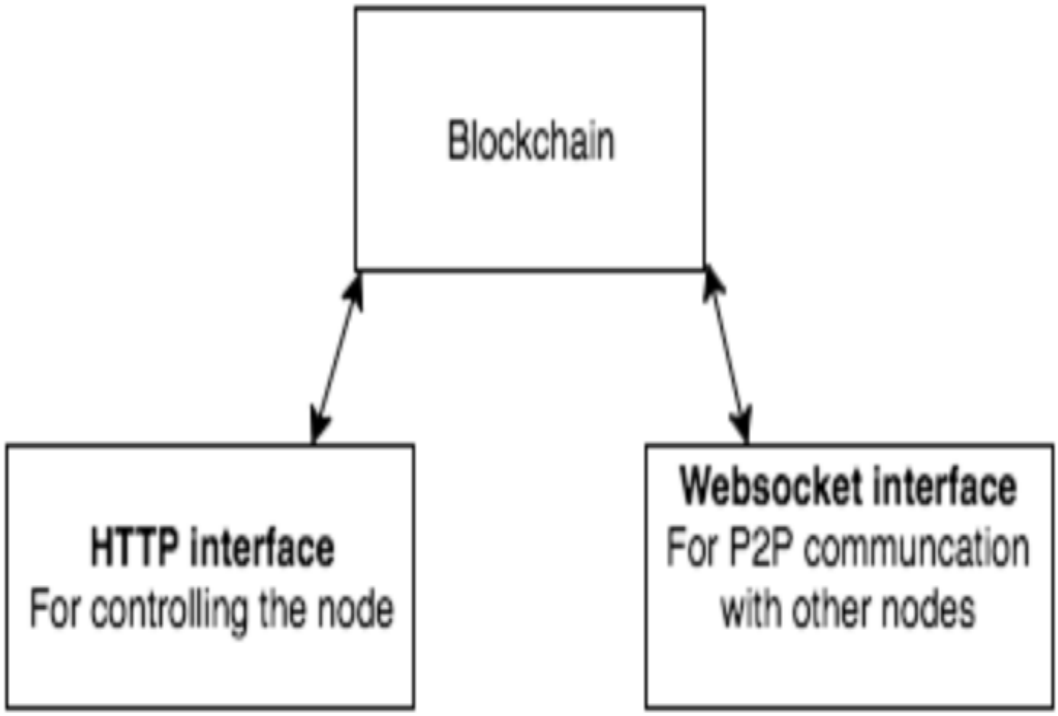


Рис 4.11. - Основні елементи мережевої передачі блоків

Після реалізації NaiveChain стає можливим передача захищених даних від одного користувача до іншого в мережі.

Додаємо інформацію що потрібно передати користувачу в мережі, та вказуємо його IP адресу, якій надаємо доступ:

var say = "Hello to another user"

allow ip = '176.37.3.213'

log = '176.37.3.192'

user log = admin

user pass = 000000

Після вказівок для блоків та вузлів в блокчейн, створюємо сайт з логіном, полем для відправки повідомлення, датою, часом та іменем відправника повідомлення, для зручного обмілу повідомленнями.

Сайт для логіну буде створено за допомогою фреймворку Django, куди буде відправлятися зашифрована за допомогою блокчейн технологій текстова інформація.

Отже, для початку потрібно створити поля для логіну до панелі обміну повідомленнями:

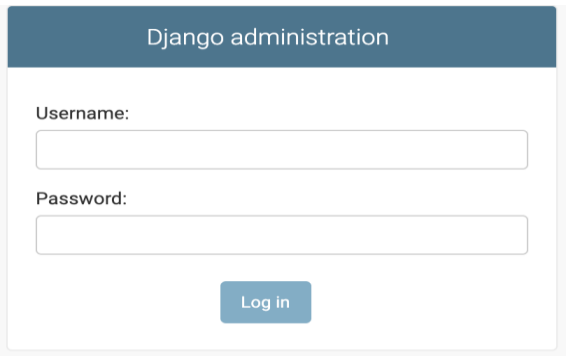


Рис. 4.12. - Панель логіну до системи передачі тексту

Після логіну користувач потрапляє до панелі, через яку, отримавши певний доступ, може безпечно передавати інформацію іншому користувачеві мережі:

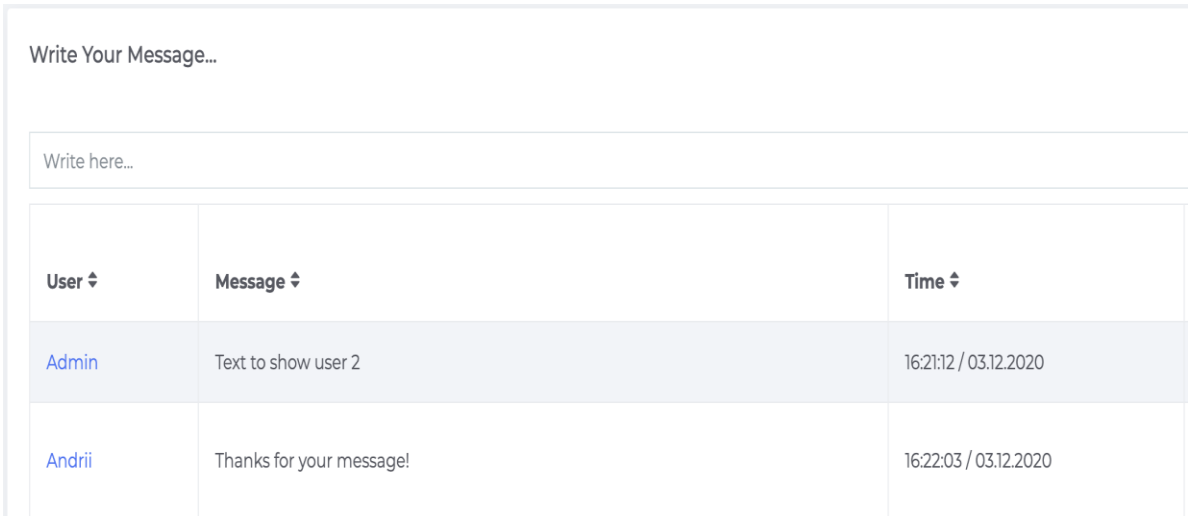


Рис. 4.13. - Захищена передача текстових даних за допомогою блокчейн

Ці повідомлення можливо побачити користувачеві з певною IP адресою, якій ми надали доступ через блоки та вузли блокчейн. Текстова інформація є захищенною завдяки методам шифрування блокчейн технологій. Користувачі яким

не надали певний доступ для логіну в панель обміну повідомленнями, не зможуть потрапити до даного сайту.

В панелі можна побачити користувача, що відправляє повідомлення, поле для набору текста повідомлення, текст повідомлення, дату та час відправлення повідомлення.

Висновок до розділу 4

Було розроблено методи захисту текстової інформації задопомогою блокчейн технологій, побудовано блоки, хеш, згенеровано ключі доступу, створено панель на базі фреймворку Django для обміну захищеними даними між користувачами в мережі. В результаті, було створено працюючу панель обміну захищенними даними в мережі за допомогою блокчейн технологій.

ВИСНОВОК

Провівши та проаналізувавши роботу над методами захисту текстової інформації за допомогою блокчейн технологій, можна сказати, що розвиток сучасних технологій надає змогу вдосконалювати та оптимізувати процеси захисту текстової інформації при передачі між користувачами в мережі. Також, дані методи є ефективними в роботі багатьох сфер сучасних технологій, коммунікацій, фінансів. Дана тема, є актуальною як для звичайних користувачів мережі, так і для великих компаній, які зацікавлені в цілісності та захищеності своїх даних в мережі.

Основним завданням дослідження є оптимізація та розробка методів шифрування текстової інформації за допомогою блокчейн технологій. В роботі представлені види та методи шифрування, способи їх практичного використання.

Було опрацьовано теоретичний матеріал, обрано інструментальні засоби, викладено теоретичні основи та суть досліджуваної проблеми, визначенні основні завдання проекту, також був проведений огляд літератури присвяченої темі та питанням що розглядаються.

Описано процес роботи з методами шифрування текстової інформації за допомогою блокчейн технологій, та їх використання на практичному прикладі.

У першому розділі описано основні методи та алгоритми шифрування текстової інформації, на яких базується робота з блокчейн технологіями, їх різновиди та методи використання на практиці.

В другому розділі мова йде про математичну базу, на якій побудовані блокчейн технології, яка допомагає оптимізації процесів та універсальності роботи з ними.

Третій розділ наглядно показує архітектуру блокчейн технологій, яка впливає на цілісність та захищеність текстової інформації що буде передаватись в мережі. Та показує основні складові архітектури блокчейн, такі як: структури передачі даних в блокчейн для шифрування текстової інформації, побудові блоків в блокчейн, пов’язування блоків у блокчейні, локалізація блокчейн.

Четвертий, заключний розділ показує безпосередньо розробку та використання на практиці методи шифрування текстової інформації за допомогою блокчейн технологій, та побудову сайту для обміну захищенною текстовою інформацією між користувачами в мережі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Документація Blockchain Platform for the Enterprise [Електронний ресурс]. — Режим доступу: https://hyperledger-fabric.readthedocs.io/en/release-2.0/
2. Документація Cryptography [Електронний ресурс]. — Режим доступу: https://en.wikipedia.org/wiki/Cryptography
3. Блог компанії IBM про технології блокчейн [Електронний ресурс]. — Режим доступу: https://www.ibm.com/blogs/blockchain/
4. Quora, розділ про розвиток технологій блокчейн [Електронний ресурс]. — Режим доступу: https://www.quora.com/q/cryptoblockchain
5. Don T., Alex T., Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World — Kindle Edition; 2016.
6. Jean-Philippe A., Serious Cryptography: A Practical Introduction to Modern Encryption — No Starch Press, 2017.
7. Блокчейн для бізнесу // У. Могайар, В. Бутерін. – М.: Ескімо 2018. – 224 с.
8. Блокчейн: план нової економіки // М. Сван. – O'Reilly 2015. – 130 с.
9. Освоєння біткойнів: розблокування цифрових криптовалют // М. Антонопулос – O'Reilly 2014. – 298 с.
10. Значення веб // К. Скіннер – MCI 2016. – 424 с.
11. Наука блокчейну // Р. Ваттерлофер – CreateSpace 2016. – 123 с.
12. Блокчейн контракти та кібер-закони // П. Дуггал – CreateSpace 2019. – 55 с.
13. Стандарти блокчейну та ассети // С. Аммус – HardCover 2018. – 304 с.
14. Блокчейн повітряна куля або революція: сьогодення і майбутнє блокчейну // Н. Мехта, А. Агаши. – Paravane Ventures 2019. – 331 с.
15. Пояснення технології блокчейн // А. Норман – CreateSpace 2017. – 126 с.

ДОДАТОК

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ/ НАВЧАЛЬНО НАУКОВИЙ ІНСТИТУТ ДЕННОЇ ФОРМИ НАВЧАННЯ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Методи шифрування текстової інформації за допомогою блокчейн технології

Виконав: Дзима Андрій

Студент 6 курсу, групи ПДМ-61

Керівник: кандидат т.н. доцент Щербина І. С.

Київ – 2020

1

Об’єкт, предмет, мета дослідження

Об’єкт дослідження:
Методи шифрування текстової інформації

Предмет дослідження:
Блокчейн технології

Мета дослідження:
Захист та шифрування текстової інформації за допомогою блокчейн технологій

2

Актуальність:

В час стрімкого розвитку мережі інтернет все частіше постає питання безпечної передачі текстової інформації між користувачами. Пересилання текстової інформації через мережу інтернет - поширена ситуація, а захист таких даних відіграє дуже важливу роль в функціонуванні великої кількості компаній. На даний час, існує ряд варіантів передачі текстової інформації, які потребують належного рівня захисту в процесі передачі. Методи передачі та шифрування залежать від загальних потреб відправника та отримувача.

Роль блокчейн технологій в шифруванні текстової інформації

- забезпечення цілісності даних що передаються;
- забезпечення захищеності даних що передаються;
- зручність роботи з великими об’ємами даних;
- швидкість передачі зашифрованих даних.

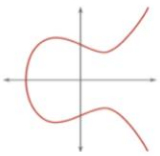
Алгоритми блокчейн технологій, що забезпечують функціонування шифрування текстової інформації:

Блокчейн алгоритми шифрування:

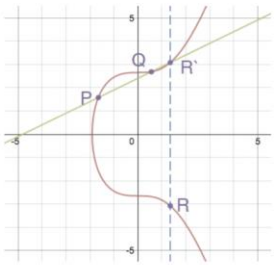
1. Симетричне шифрування
2. Алгоритм DES
3. Алгоритм AES
4. Асиметричне шифрування
5. Алгоритм Діффі-Хеллмана
6. Алгоритм з відкритим ключем RSA

Математична складова блокчейн методів шифрування

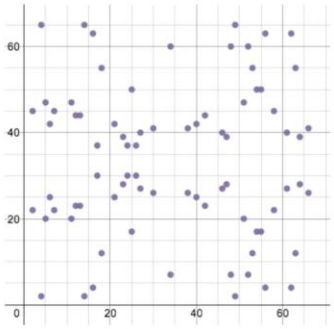
Еліптична крива:



Еліптична крива для побудови блоків:



Еліптична крива, що вказує на росташування блоків:



Поняття архітектури блокчейн

Архітектура блокчейну складається з таких елементів, як вузол - користувач або комп'ютер, який має повну копію блокчейну, блок - структура даних, що використовується для ведення набору транзакцій, і транзакція - найменший блок системи блокчейну (який містить записи , інформацію тощо). Всі блоки в певному порядку розташовані в ланцюжок, які допомагають перевірити кожен блок. Процес всередині системи організований відповідно до протоколу.

Розробка архітекрути блокчейн технологій, котра складається з:

- Логіки блокових елементів
- Компоненти шифрування інформації
- Структура передачі даних
- Побудова блоків
- Побудова блоку заголовку
- Повязування блоків
- Доведення транзакцій у блоках
- Локалізації

Складові методів блокчейн технологій для шифрування

Розробка хеш-блоку для шифрування текстової інформації

Генерація блоку для передачі інформації кінцевому користувачу

Зберігання та перевірка цілісності блоків блокчейн

Формування ланцюгів між блоками в блокчейн

Зв'язок між блочними елементами

Налаштування зв’язку з http-сервером

Реалізація методів блокчейн шифрування

Було створено захищене середовище(панель) для обміну текстовою інформацією в мережі між користувачами, доступ до якого надавалось по IP адресі.

Панель була створена на основі фреймворку Django.

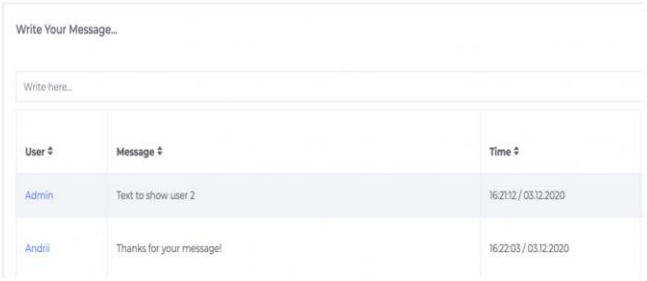
Користувачі мережі, які не мають доступу по IP адресі не можуть потрапити до панелі обміну текстової інформації.

Створення панелі для обміну текстової інформації

Було розроблено панель для логіну та обміну інформацією за допомогою фреймворку Django

Панель логіну:

Панель обміну інформацією:



Висновки та апробація

Провівши та проаналізувавши роботу над методами захисту текстової інформації за допомогою блокчейн технологій, можна сказати, що розвиток сучасних технологій надає змогу вдосконалювати та оптимізувати процеси захисту текстової інформації при передачі між користувачами в мережі.

Також, дані методи є ефективними в роботі багатьох сфер сучасних технологій, коммунікацій, фінансів. Дана тема, є актуальною як для звичайних користувачів мережі, так і для великих компаній, які зацікавлені в цілісності та захищеності своїх даних в мережі.

Основним завданням дослідження є оптимізація та розробка методів шифрування текстової інформації за допомогою блокчейн технологій.

В роботі представлені види та методи шифрування, способи їх практичного використання. Було опрацьовано теоретичний матеріал, обрано інструментальні засоби, викладено теоретичні основи та суть досліджуваної проблеми, визначенні основні завдання проекту, також був проведений огляд літератури присвяченої темі та питанням що розглядаються. Описано процес роботи з методами шифрування текстової інформації за допомогою блокчейн технологій, та їх використання на практичному прикладі.

Апробація результатів. Основні положення в результати магістерської роботи доповідались і обговорювались на науково-технічній конференції. Дзима А.В. Методи шифрування текстової інформації за допомогою блокчейн технологій // Загально-університетська, науково-технічна конференція «Проблеми комп'ютерної інженерії»; від 02.12.2020 р. За результатами роботи опублікована стаття А.В. Дзима, І.С. Щербина «Методи шифрування текстової інформації за допомогою блокчейн технологій» , Журнал "Зв'язок", Випуск №6, 2020, Київ.

ДЯКУЮ ЗА УВАГУ!