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

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
_____ С.В. Казмірчук
«_____»_____ 20__ р.

На правах рукопису
УДК 004.45

ДИПЛОМНА РОБОТА

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»

Тема: Система забезпечення цілісності електронних документів на основі технології блокчейн

Виконавець: Р.В. Ворса
Керівник: к.т.н., доцент О.М. Кулініч
Нормоконтролер: к.т.н., доцент О.М. Кулініч

Київ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет: Кібербезпеки, комп'ютерної та програмної інженерії

Кафедра: Комп'ютеризованих систем захисту інформації

Освітній ступінь: Бакалавр

Спеціальність: 125 «Кібербезпека»

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ С.В. Казмірчук

«__» _____ 20__ р.

ЗАВДАННЯ

на виконання дипломної роботи

здобувача вищої освіти Ворси Руслана Валерійовича

1. Тема: *Система забезпечення цілісності електронних документів на основі технології блокчейн*

затверджена наказом ректора від «26» квітня 2021 р. № 652/ст.

2. Термін виконання: з 10.05.2021 р. по 20.06.2021 р.

3. Вихідні дані: проаналізувати основні поняття захисту інформації технології блокчейн; на основі аналізу виділити особливості технології блокчейн; дослідити властивості ефективної системи електронного документообігу; розробити модель, алгоритм та програмне забезпечення системи цілісності електронних документів на основі технології блокчейн.

4. Зміст пояснювальної записки: аналіз складових захисту та можливостей інформації технології блокчейн; аналіз можливостей технології блокчейн для побудови системи електронного документообігу; розробка та тестування програмного забезпечення запропонованої системи.

КАЛЕНДАРНИЙ ПЛАН

виконання дипломної роботи

№ п/п	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1.	Уточнення постановки задачі	19.04.2021	*Виконано*
2.	Аналіз літературних джерел	20.04.2021	*Виконано*
3.	Обґрунтування вибору рішення	21.04.2021	*Виконано*
4.	Збір інформації	22.04.2021	*Виконано*
5.	Дослідження основних складових захисту інформації технології блокчейн	08.05.2021	*Виконано*
6.	Аналіз особливостей технології блокчейн	11.05.2021	*Виконано*
7.	Дослідження ефективної системи електронного документообігу	14.05.2021	*Виконано*
8.	Розробка алгоритму та програмного забезпечення системи цілісності електронних документів на основі технології блокчейн	20.05.2021	*Виконано*
9.	Перевірка на антиплагіат	05.06.2021	*Виконано*
10.	Оформлення і друк пояснювальної записки	08.06.2021	*Виконано*
11.	Оформлення презентації	09.06.2021	*Виконано*
12.	Отримання рецензій від рецензента	10.06.2021	*Виконано*

Здобувач вищої освіти Р. Ворса

(підпис, дата)

Керівник дипломної роботи О. Кулініч

(підпис, дата)

РЕФЕРАТ

Дипломна робота складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел, додатку і має 70 сторінок основного тексту, 20 рисунків, 14 сторінок додатку. Список використаних джерел містить 22 найменування і займає 2 сторінки. Загальний обсяг роботи 94 сторінки.

Метою роботи є створення системи забезпечення цілісності електронних документів на основі технології блокчейн.

В роботі розглянуто основні поняття захисту інформації технології блокчейн, звернено увагу на значимість інформаційної безпеки, розглянуто особливості асиметричної криптографії, що є основою технології блокчейн. Наведено визначення, класифікацію та особливості технології блокчейн, особливості застосування цифрового підпису та хешування, яке забезпечує незмінність всього ланцюжка транзакцій блокчейн. Вказані основні структурні блоки та алгоритми досягнення консенсусу.

В роботі описано логічну структуру та основні компоненти, що необхідні для реалізації системи, наведено програмну реалізацію системи електронного документообігу з накладеним цифровим підписом на основі технології блокчейн.

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

Можливі напрямки розвитку цієї роботи пов’язані із застосовування в подальшому для розробки та впровадження системи електронного документообігу в межах організації.

Ключові слова: блокчейн, інформаційна безпека, криптографічні системи, система електронного документообігу, електронний документ, цифровий підпис, технологія блокчейн.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БСШ – блоковий симетричний шифр

ЕД – електронний документ

ЕДО – електронний документообіг

ЕЦП – електронно цифровий підпис

ЗІ – захист інформації

ІТС – інформаційно-телекомунікаційна система

КЗІ – криптографічний захист інформації

НСД – несанкціонований доступ

СЕД – система електронного документообігу

ТЗІ – технічний захист інформації

ЦП – цифровий підпис

ВСТУП

Актуальність роботи полягає у швидкому розвитку технологій інформаційного суспільства та зростаючому обсязі документообігу, у необхідності електронної системи управління документами, яка повинна функціонувати, щоб користувачі могли постійно отримувати доступ та перевіряти документи в електронному вигляді. Така система на сучасному підприємстві є нагальною потребою. За мінімальних витрат на технічне обладнання та програмне забезпечення впровадження системи електронного документообігу може суттєво покращити якість та продуктивність роботи з різними документами та полегшити перегляд інформації. Запровадження системи електронного документообігу забезпечить абсолютну прозорість, безпеку, цілісність та доступність, економічну ефективність, усунення ризиків втрати документів та несанкціонований доступ до інформації. У свою чергу, технологія блокчейн забезпечує безпечне та функціональне рішення, яке дозволяє налагодити безперебійну роботу з важливими документами в межах державної установи. Користувачі такого рішення забезпечуються надійним та безпечним робочим середовищем, де ніхто не може впливати на надійність даних, а також навмисно їх фіксувати. Звідси постає актуальне питання реалізації системи електронного документообігу надійного зберігання файлів, яка дозволить забезпечити цілісність та доступність документів.

Мета роботи: впровадження системи електронного документообігу на підставі технології блокчейн.

Для реалізації цієї мети в роботі вирішуються такі завдання:

- розглянути основні компоненти захисту інформації технології блокчейн (РОЗДІЛ 1);
- дослідити особливості та принципи роботи технології блокчейн, що необхідно для реалізації практичного завдання (РОЗДІЛ 2);

- розглянути принципи та особливості продуктивної системи електронного документообігу та розібрати можливості технології blockchain для її побудови (РОЗДІЛ 3);
- реалізувати систему електронного документообігу за допомогою технології blockchain з накладеним цифровим підписом (РОЗДІЛ 4).

Об’єкт дослідження: зберігання електронних документів, що містять цифрові підписи, з використанням технології шифрування та блокчейну.

Предмет дослідження: система електронного документообігу на основі технології блокчейну.

При написанні роботи використовувалися застосування методів аналізу та синтезу для використання елементів технології blockchain для реалізації системи електронного документообігу. Використання методу моделювання для створення ефективної моделі системи електронного документообігу. Програмний засіб реалізований на мові PHP та JavaScript. Для проведення реалізації програмного засобу застосовувалося програмне середовище PhpStorm.

Основним результатом роботи є реалізована система електронного документообігу з використанням технології blockchain з можливістю накладання цифрового підпису.

РОЗДІЛ 1. ОСНОВНІ СКЛАДОВІ ЗАХИСТУ ІНФОРМАЦІЇ ТЕХНОЛОГІЇ БЛОКЧЕЙН

В даному розділі наведено визначення інформації та її властивості, визначено сутність криптографії та вказано на особливості асиметричної криптографії, що дозволяє забезпечити захист даних користувача технології блокчейн. Також вказано на особливості застосування цифрових підписів, що забезпечує цілісність даних та наведено особливості хешування, яке забезпечує незворотність всього ланцюжка транзакцій, що є основою блокчейн.

1.1 Основні поняття захисту інформації

Інформаційний термін походить від англійського слова «Informatio», що означає роз'яснення, повідомлення, виклад. З точки зору знань філософії є зображення наявного навколо за допомогою даних.

Термін «Інформація» налічує велику кількість визначень, у свою чергу ми будемо використовувати наступне:

Інформація – це документально або публічно розкрита інформація про явища та події, що виникають у громаді, державі, та навколишньому середовищі [1].

У науковій літературі термін «Інформація» має інше поняття. Інформація – це інформація про людей, речі, технології, інструменти, ресурси, події та явища, що здійснюються в усіх сферах операцій країни, життя громади та довкілля, не залежачи від форми положень. Інформація може бути представлена у вигляді сигналів, зображень, знаків, або в інший спосіб [2].

Ключові особливості інформації включають:

- конфіденційність – особливість інформації бути захищеною від несанкціонованого доступу;
- цілісність – риса інформації бути захищеною від несанкціонованого спотворення, руйнування або знищення;
- доступність – вміння інформації бути захищеною від несанкціонованого блокування;
- цілісність – спроможність інформації об’єктивно зображати події що трапляються в Всесвіті.

Всі сфери людського життя безпосередньо пов’язані з джерелом інформації, яке має цінність і вимагає захисту від різних впливів, які можуть призвести до руйнування або порушення цілісності. Сьогодні потреба в стабільному та постійному захисті інформації постійно зростає.

Надійне збереження інформації має велику цінність у захисті інтересів країни. Умовою розвивання суспільства та країни є налаштування цивілізованого та безпечного інформаційного середовища. Останнім часом у світі виникає інноваційний перехід в процесах управління, завдяки успішному впровадженню сучасних технічних розробок. Одночасно зростає небезпека зовнішнього втручання в процес інформаційних систем, і серйозність наслідків несанкціонованого втручання значно підросла. Внаслідок цього проблема охопила велику кількість уваги від багатьох країн світу та пошуку її надійних рішень. Однією з головних ознак науково-технічного прогресу сьогодні є стрімкий розвиток інформаційних технологій. Однак із широким використанням цих технологій у повсякденному житті та в професійній галузі виникає проблема безпеки цих технологій.

Інформаційна безпека (згідно із законодавством України) – це питання безпеки інформаційного середовища громади, що дає можливість утворення, та застосування процесу розвитку в інтересах суспільства, підприємств, країни [3].

Об’єктами безпеки інформаційного середовища є людина, суспільство та держава, обов’язком інформаційної безпеки є забезпечення їхніх інтересів.

Захист інформації – це процес, спрямований на уникнення викриття захищеної інформації, несанкціонований та ненавмисний вплив на інформацію що захищається. Встановлення системи захисту інформації є метою інформаційної безпеки підприємства.

Стрімки розвиток інформаційних технологій призвів до нових розробок в галузі інформаційної безпеки, що має вирішальне значення для сучасного суспільства. Проблеми розробки та впровадження методів захисту інформації стосуються не тільки стеганографії та криптографії, а й майже кожної науки, завдяки високому рівню автоматизації різноманітних ділянок людської діяльності. Серед усіх чинних методів, криптографічні займають перше місце в захисті даних від несанкціонованого втручання. Не пов’язані з другими методами тим, що вони покладаються виключно на особливість інформації і не застосовують властивості матеріального середовища, характеристики одиниць обробки.

Криптографія – наука про шифрування – займає особливу роль в сучасній обчислювальній інфраструктурі. Шифр (Криптографічні алгоритми) широко застосовується для захисту особистих даних користувачів, зберігання паролів, виявлення помилок в масивах відомостей, тестування логічних пристроїв, а також застосовується в вирішенні інших завдань. Криптографія забезпечує спосіб безпеки інформації й, отже, є частиною заходів захисту інформації [4].

Приховати (або відновити) зміст інформації, перевірка її справжності, цілісності, авторства тощо є особливістю Криптографічного захисту. Цей тип захисту, реалізований шляхом перетворення інформації за допомогою приватних даних [5].

Ціль криптографії полягає в забезпеченні приватності даних (чистий текст, секретний ключ) за допомогою їх шифрування. На додаток до збереження приватності, використовується для розв’язання таких проблем, як:

– автентифікація (ідентичність), тобто одержувач може ідентифікувати відправника, а злочинець не може сховатися під ним;

– цілісність означає, що одержувач може перевірити несанкціоновану зміну тексту, а злочинець не може передати фальшивий текст, як ніби справжній;
– відмова від авторства означає, що відправник більше не може відмовити у поданні даних.

Нині криптографія включає чотири основні частини:

– симетричні системи шифрування (приватний ключ або системи з одним ключем);
– асиметричні системи шифрування (відкритий ключ або системи з двома ключами);
– абстрактний чи конкретний протокол;
– управління ключами.

Діяльність системи блокчейн забезпечує криптографія. Ґрунтуючись на принципах математичної та економічної технологій архітектури блокчейну, мережа передбачає цю довіру серед учасників. Криптографія крім того, забезпечує захист на основі прозорості та можливості огляду всіх транзакцій.

Маскувальні методи шифрування – шифрування або інші методи перетворення інформації, внаслідок чого вміст стає недоступним без надання зашифрованого ключа та зворотного перетворення [6].

Найнадійнішим способом захисту безперечно є криптографічний метод, оскільки він безпосередньо захищає інформацію замість доступу.

Сучасні криптографічні системи захисту інформації мають такі вимоги:

– зашифрований текст повинен бути читабельним лише одним ключем;
– кількість транзакцій, необхідних для визначення ключа шифрування, що використовується для частини зашифрованого повідомлення та відповідного простого тексту, не повинно бути менше загальної кількості можливих ключів;
– інформація алгоритму шифрування не повинна впливати на надійність захисту;

- невелика зміна ключа повинна призвести до значної зміни типу зашифрованого повідомлення, навіть якщо використовується той самий ключ;
- структурні елементи алгоритму шифрування повинні бути незмінними;
- додаткові біти, введені в повідомлення в процесі шифрування, повинні бути повністю та надійно заховані в зашифрованому повідомленні;
- довжина зашифрованого тексту повинна дорівнювати довжині вихідного тексту;
- між ключами, які постійно використовуються в процесі шифрування, не повинно бути простих залежностей;
- він повинен забезпечувати надійний захист будь-якої ключової інформації з набору можливих;

Відмінні методи шифрування, забезпечують безперервність огляду транзакцій блокчейну, проблеми автентифікації блокчейну в цілому, а також контролюють доступ до системи та вирішують завдання автентифікації.

Блокчейн, тип шифрування, що використовується при шифруванні відкритим ключем, набагато краще підходить для функцій, пов’язаних із цією технологією, ніж симетричне шифрування ключів. Шифрування відкритим ключем, також відоме як асиметрична криптографія, є стандартним інструментом оптимізації шифрування симетричного ключа, оскільки він надає можливість передавати інформацію за використанням відкритого ключа, яким можна поділитися з усіма. Шифрування відкритим ключем – це математична основа для комп’ютерної та інформаційної безпеки. На захист даних користувача виступають асиметричні алгоритми, що передаються в мережу блокчейн.

Асиметричні системи шифрування є дійовими системами збереження криптографічних відомостей. Алгоритми відкритого ключа спроєктовані таким чином, щоб ключ дешифрування ключа, який використовується для шифрування, був іншим. Крім того, мінімум продовж певного проміжку часу, ключ розшифрування неможливо обчислити за ключем шифрування. Ключ

шифрування може бути відкритим, і кожен може використовувати відкритий текст для шифрування (несекретного прийому каналу), але тільки власник відповідного ключа дешифрування може розшифрувати повідомлення, оскільки їх алгоритм називається «з відкритим ключем». У відповідних системах ключ шифрування іменують як відкритий, а ключ дешифрування - закритий. Один з найбільш розповсюджених алгоритмів систем з відкритим ключем, який сьогодні використовується, відомий як RSA.

Розшифрувати інформацію за допомогою відкритого ключа неможливо. Для дешифрування даних користувач використовує секретний (приватний) другий ключ. Ключ дешифрування неможливо визначити за ключем шифрування. Шляхом відкритого публічного ключа і підпису, всі вузли мережі можуть перевірити і прийняти транзакцію за дійсну, підтверджуючи, що особа є насправді власником. Схема передачі даних в асиметричних криптосистемах зображена на рис.1.1.

Отримувач сповіщає свій відкритий ключ, який допускає зашифрувати повідомлення для нього. Приватний ключ відомий лише одержувачу повідомлення. Коли комусь потрібно надіслати зашифроване повідомлення, вони зашифровують його за допомогою відкритого ключа одержувача. Отримавши повідомлення, він розшифровує повідомлення своїм приватним ключем.

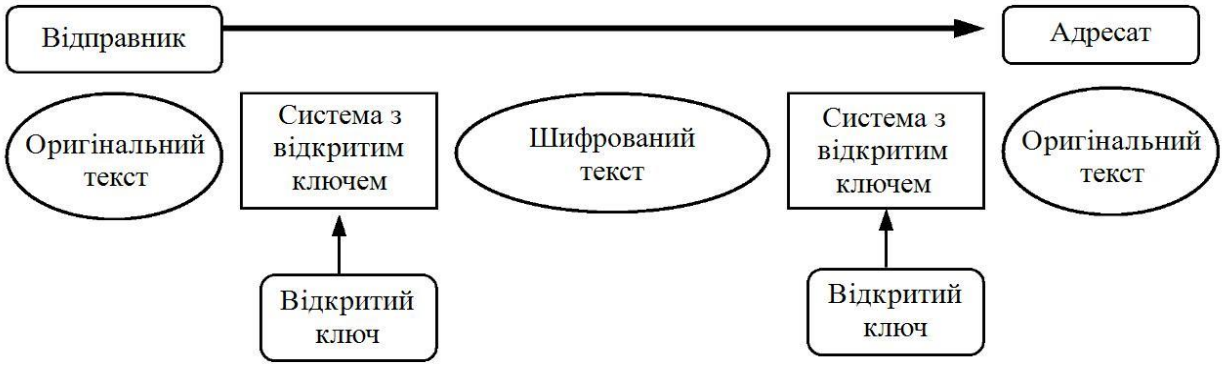


Рис. 1.1. Схема передачі даних в асиметричних криптосистемах

Для забезпечення надійного захисту інформації повинні бути системи відкритих ключів (СВК) що дотримуються двох чітких і важливих правил:

- перетворення вихідного тексту не буде оборотним і не повинно призвести до повторення інформації, зашифрованої за допомогою відкритого ключа;
- вказівка приватного ключа на основі відкритого ключа повинна бути неможливою, враховуючи сучасні досягнення та можливості обчислювальної техніки. При цьому, обов’язковою є точна оцінка складності (кількості операцій та часу) для зламу шифру.

Концепція шифрування відкритим ключем пов'язана з односторонніми функціями або функціями f (x); Знання значення аргументу "x" досить просто, щоб знайти значення функції, але теоретично досить важко визначити аргумент із функції. Отже, насправді, щоб знайти аргумент функції, користувач повинен мати додатковий спосіб полегшити розшифровку та знайти спосіб легкого відтворення вихідного значення. Таким чином, користувацький ключ рухається і в цьому прикладі значення функції є таким, що $f(x) = y$, де ‘y’ – закритий ключ в системі СВК.

Як правило, усі новітні системи шифрування з відкритим ключем використовують один із таких типів незворотного перетворення:

- ділення великих чисел на прості множники;
- обчислення значень логарифмічної функції в межах обмеженого простору;
- обчислення коренів алгебраїчних рівнянь.

Також, говорячи про практичне значення СВК, слід відзначити наступні алгоритми для можливих реалізацій:

- як незалежні ресурси захисту передачі та зберігання даних;
- як середовище розподілу ключів – алгоритми СВК підходять для розподілу ключів, які є незначними у поєднанні з іншими традиційними системами шифрування, і обсяг інформації є досить складним порівняно на практиці за допомогою СВК;

– як засіб автентифікації користувачів – системи з відкритим ключем можуть використовувати найрізноманітніші криптоалгоритми (RSA, Ель-Гамаля або Діффі-Хелмана та криптосистеми на основі еліптичних рівнянь).

Фактично, вибір алгоритму лише визначає спосіб та складність шифрування ключа, а не принципову різницю між методом передачі інформаційних потоків.

1.2 Використання цифрового підпису в блокчейн

Важливим аспектом в захисті інформації за допомогою криптографії з відкритим ключем є створення цифрового підпису, який забезпечує цілісність даних. Це робиться шляхом об’єднання закритого ключа користувача з даними, які він має намір підписати, за допомогою математичного алгоритму. ЕЦП ґрунтується на використанні асиметричного шифрування та хеш- функціях.

Оскільки фактичні дані самі по собі є частиною цифрового підпису, мережа не розпізнає їх як дійсні, якщо будь-яка їх частина підроблена. Редагування навіть найменшого фрагмента даних змінює форму всього підпису, роблячи його помилковим і застарілим. Завдяки цьому технологія блокчейн здатна гарантувати, що будь-які дані, що записуються, є істинними, точними і незміненими. Цифрові підписи – це те, що надає записаним в блокчейн даним незмінність.

Електронно-цифровий підпис – це послідовність байтів, яка формується шляхом перетворення підписування інформації за криптографічним алгоритмом і призначена для перевірки авторства ЕД. Цифрові підписи є одним з основних факторів забезпечення безпеки і цілісності даних, які записуються в блокчейн. Вони є стандартною частиною більшості протоколів блокчейн, в основному використовуються для захисту транзакцій і блоків транзакцій, передачі конфіденційної інформації, поширення програмного забезпечення,

управління контрактами та будь-яких інших випадків, коли важливо виявити і запобігти будь-яке зовнішнє втручання. Цифрові підписи використовують асиметричну криптографію, що означає, що інформацією можна ділитися з ким завгодно за допомогою відкритого ключа [7].

Цифрові підписи забезпечують три ключові переваги зберігання і передачі інформації в блокчейн. Перш за все, вони гарантують цілісність. Теоретично зашифровані дані, які передаються, можуть бути змінені непомітно хакером.

Однак, якщо це станеться, підпис зміниться, ставши неправильним. Тому дані з ЦП не тільки в безпеці від перегляду, але і дають можливість виявити, чи були вони підроблені, зміцнюючи тим самим їх незмінність. Цифрові підписи не тільки захищають дані, але і особу відправника. Володіння ЦП завжди пов’язане з певним користувачем, тому можна бути впевненим, що вони спілкуються з тим, з ким мають намір це робити.

Цифрові підписи в блокчейн базуються на криптографії з відкритим ключем (див. рис. 1.2).У ній використовуються два ключа. Перший – закритий ключ, який потрібен для формування цифрових підписів і зберігається в секреті. Другий – відкритий ключ, який використовується для перевірки електронного підпису, сформованого секретним ключем. Відкритий ключ насправді можна обчислити на основі приватного ключа, але для зворотного перетворення потрібна обчислювально неможлива сума, порівнянна з брут-форсом.

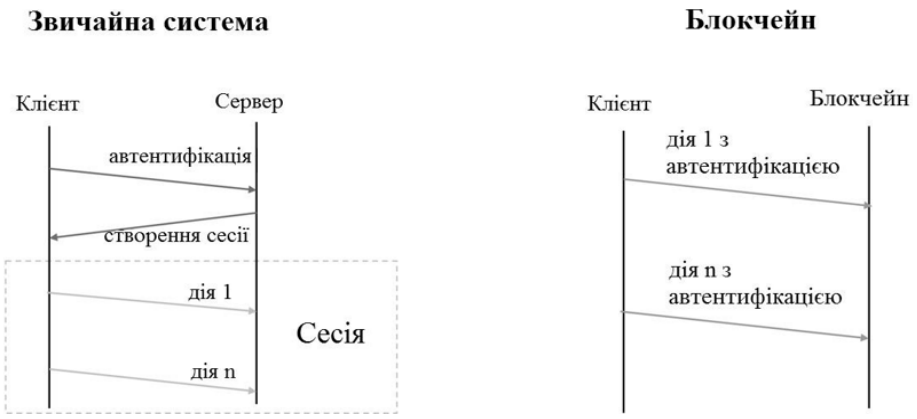


Рис. 1.2. Порівняння традиційної системи та технології блокчейну

Щоб перевірити, що повідомлення чи інформація дійсно належить члену мережі, повідомленням слід присвоїти так звані цифрові підписи.

Цифровий підпис або електронний підпис (англ. signature – підпис) – це рядок символів, що залежить як від відправника так і від змісту повідомлення (див. рис. 1.3).

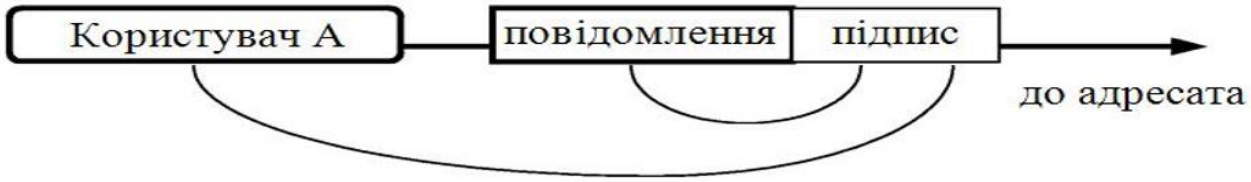


Рис. 1.3. Цифровий підпис як частина повідомлення

Жоден член мережі, крім Користувача А (відправник), не може вказати формат підпису для кожного конкретного повідомлення. Ніхто, включаючи певного юзера, не має змоги вносити зміст повідомлення так, щоб сигнатура залишилась незмінною. Однак одержувач повідомлення повинен мати можливість перевірити підпис відправника. Для перевірки дійсності цифрового підпису користувач В (одержувач) повинен надати сторонній особі С інформацію (мережу або сервер перевірки підпису) інформацію про те, які дані використовуються для перевірки підпису. Якщо повідомлення надсилається безпосередньо від відправника одержувачу, який не є третьою стороною, це "оригінальний цифровий підпис".

Вищевказана модель має ряд недоліків:

- мережа припускає існування третьої сторони - клієнта, якому однаково довіряють як відправник, так і одержувач;
- відправник, одержувач та клієнт автентифікації повинні обмінятися значним обсягом службової інформації до того, як буде доставлено саме повідомлення;
- передача даної інформації повинна відбуватися у закритій формі, тому її використання неефективне.

Однак навіть така схема цифрового підпису успішно використовується в цифрових системах, де необхідно дотримуватися двох простих правил: обов'язкове шифрування повідомлень, що надсилаються по мережі для автентифікації й розпізнавання інформації.

Існує безліч різних схем криптографії з відкритим ключем. Одним з таких алгоритмів може бути RSA (див. рис. 1.4). Вибір асиметричного шифрування довів той факт, що інші учасники мережі повинні переконатися, що власник блоку вніс зміни і блок підписаний своїм підписом. Дві найпопулярніші схеми з них – це схеми на основі розкладання на множники (RSA) і схеми на основі еліптичних кривих. Останні більш популярні в блокчейн через менший розмір ключів і підписів.

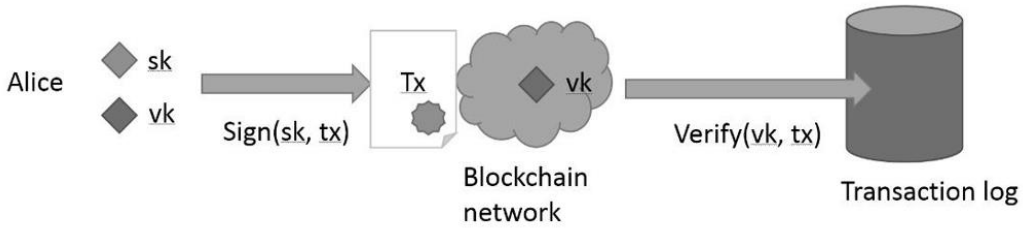


Рис. 1.4. Схема роботи шифрування в блокчейн

Закритий (приватний) ключ генерується самим користувачем, використовується для підпису транзакцій. Особа, що має приватний ключ, може бути представлена доступом до осередку блокчейну бази даних, який зберігається в таємниці.

Відкритий ключ повинен створюватися на основі приватного ключа, тобто між ними існує математична залежність. Більше того, блокчейн використовується як адреса цього блоку, а також інформація в інших блоках може бути опублікована третіми сторонами для перевірки справжності підпису. Інформація про відкритий ключ не дозволяє їй визначати приватний ключ.

Для створення підпису потрібно:

– асиметричний алгоритм шифрування (наприклад, RSA);

– Хеш функція (наприклад, SHA-512);
– інформація, що необхідно підписувати.

Оскільки асиметричні алгоритми досить повільні в порівнянні з симетричними, то обсяг підписуваних даних відіграє велику роль і якщо він великий, то зазвичай беруть хеш від підписуваних даних, а не самі дані. Хеш, SHA-512, приймає вхідну інформацію і отримується за допомогою хеш-функцій, оскільки повертається хеш певної фіксованої довжини. ЕЦП розміщується у хеш-значенні, а не в самому документі. Хеш-функція не являється частиною підпису, тому можна надійно використовувати в схемі певні хеш-функції.

Етапи:

- Створюємо пари відкритого та приватного ключів за допомогою RSA;
- Змінені дані підставляються функцією SHA-512 і отримується хеш;
- Отриманий хеш і приватний ключ обмінюються за допомогою асиметричної функції шифрування RSA, тобто RSAEncode (хеш від інформації, закритий ключ), як результат, на виході отримаємо рядок – ЕЦП.

Алгоритм підпису даних представлений на рисунку 1.5.

Сполучний хеш перераховується кожного разу, коли додається нова транзакція. Поточний блок і адреса попереднього блоку обчислюються шляхом підсумовування всіх його хеш-транзакцій:

Хеш (з'єднувач) = SHA-512 (block_prev_adress_hash + transaction_hash1 + transaction_hash2 + ... + transaction _ hashN).

Саме хеш посилання об’єднує блоки в єдиний ланцюжок і найголовніше захищає блокчейн від шахрайства зловмисників. Наприклад, якщо хтось хоче "кинути" або розмістити свій блок в середині ланцюжка, блоки, які слідують за ним, більше не пройдуть перевірку, оскільки хеші базуються на адресі, яку вони хочуть змінити або видалити.

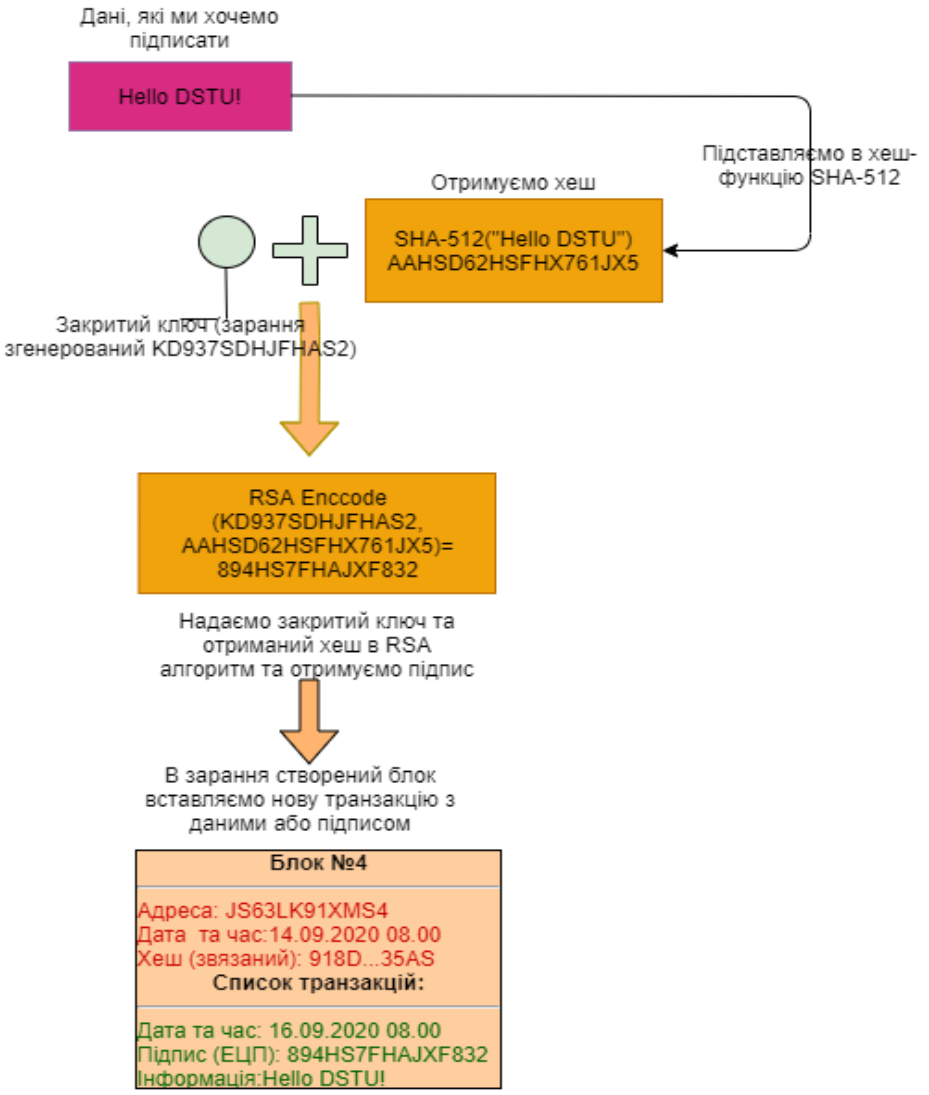


Рис. 1.5. Алгоритм підпису даних

Загальна ідея підписів з відкритим ключем виглядає наступним чином: припустимо, Аліса хоче передати біткойн Бобу. Для цього Аліса створює транзакцію, де пише, звідки взяти (вказівка на попередню транзакцію, яку Аліса отримала від когось іншого) і кому (відкритий ключ Боба) надіслати його. Аліса знає відкритий ключ Боба з інших джерел – Боб може послати його Алісі через текстовий чат або розмістити його на сайті. Аліса підписує транзакцію, застосовуючи свій приватний ключ.

Мережу біткойнів може перевірити, що будь-яка транзакція вузла була підписана певним відкритим ключем (ідентифікатором), з яким вона пов'язана, до транзакції біткойна (авторизації).

Оскільки у блокчейну відсутній центральний вузол для авторизації довільних транзакцій, безпека системи децентралізована, і ймовірність успішних спроб блокчейну зведена майже до нуля. Таким чином, блокчейн використовує цифрові підписи для автентифікації і забезпечення цілісності транзакцій (у випадку СЕД – цілісності файлів). Особливість блокчейн полягає в тому, що інформація про автентифікації "вшита" в кожну транзакцію, тому блокчейн вважається більш захищеним.

1.3 Хешування як невід'ємна складова блокчейну

Хешування - це процес перетворення масиву випадкових довжин вхідних даних у вихідний бітовий потік фіксованої довжини.

Хешування служить невід'ємною частиною блокчейну. Хеш-функції - це дані мережевого блокчейну, а хеш - це трансформація первинних даних у мережеві дані.

Застосування цифрових підписів включає використання певних функцій кодування:

$$S = H(k, T),$$

де S – підпис, k – ключ, T – оригінальний текст.

Це хеш-функція, якщо функція $H(k, T)$ задовольняє наступним умовам:

- довжина та розмір оригінального тексту мають бути довільного обсягу;
- значення $H(k, T)$ має фіксовану довжину;
- значення функції $H(k, T)$ легко обчислити для будь-якого аргументу;
- практично неможливо поновити аргумент у значенні з точки зору проектної потужності та величини прикладених сил;
- функція $H(k, T)$ – однозначна.

Однозначність поділяється на сильну і слабку. При слабкій однозначності для заданого значення Т майже неможливо знайти інший текст T', для якого $H(k, T) = H(k, T')$. При сильній однозначності для будь-якого тексту Т

неможливо знайти інший задовольняючий текст, що мав би те саме значення хеш-функції, хешуванням називають перетворення масиву випадкової довжини вхідних даних у вихідний бітовий потік фіксованої довжини. Ці типи перетворень також називаються хеш-функціями або функціями згортки, а їх результати називаються хешами, хеш-кодами, хеш-сумами або дайджестом повідомлення.

Хеш-функції використовуються для оптимізації даних, оскільки однакові значення (записи в базі даних) мають ідентичні значення хеш-функції. Цей підхід ефективний при пошуку копій для великих файлів. Криптографічна хеш-функція дозволяє перевірити, чи певні вхідні дані відповідають певному хеш-значенню, але якщо вхідні дані невідомі, здається майже неможливим відновити вхідне значення, якщо ви знаєте лише збережене значення хеш-функції. Даний механізм потрібен для перевірки та забезпечення цілісності переданих даних, а також хешу та базової одиниці для автентифікації повідомлень.

Найбільш відомими хеш-функціями є MD2/4/5 та SHA. Три алгоритми серії MD працюють за принципом перетворення тексту довільної довжини в 128-бітну сигнатуру. Вони отримали широке поширення в сучасних мережах як спосіб перевірки цілісності файлу або посилання за допомогою порівняння даних з розрахованим попередньо хешем.

Алгоритм MD5 передбачає:

- доповнення тексту до довжини, що дорівнює 448 біт по модулю 512;
- додавання довжини тексту в 64-бітному представленні;
- 512-бітні блоки повинні пройти процедуру Damgard-Merkle (цей клас перетворень передбачає розрахунок аргументів фіксованої довжини для фіксованих по довжині значень), при чому кожний блок проходить цю операцію в чотирьох різних циклах.

Стандартна хеш-функція приймає на вхід блок з певною інформацією, видаючи на виході випадкове та непередбачуване значення. Вона розроблена таким чином, що не існує оптимального та однозначно вірного методу знайти

необхідний показник без продовження перебору значень знову і знову до тих пір, доки не буде знайдено відповідний хеш-код.

З безлічі алгоритмів найпоширенішими є хеш-функції. Хеш-функції необхідні для стиснення інформації у зображення із фіксованими бітами фіксованої довжини. Хеш-функції сімейства SHA-2 користуються високою популярністю в додатках, пов'язаних із систематизацією, пошуком і захистом інформації [8].

Одним з найпопулярніших алгоритмів розрахунку блоків є саме SHA-256. Саме цим алгоритмом користується найдорожча криптовалюта у світі – Bitcoin. Причому, для підвищення рівню безпеки даний алгоритм застосовується два рази та іменується в даному випадку подвійним.

SHA-256 – це алгоритм, або, іншими словами, криптографічна хеш-функція, яка була розроблена Агентством національної безпеки Сполучених Штатів Америки. Технічні параметри SHA-256:

- показник розміру блока – 64;
- максимально допустима довжина повідомлень (байт) – 33;
- характеристика розміру дайджесту повідомлення (байт) – 32;
- стандартний розмір слова (байт) – 4;
- параметр довжини внутрішнього положення (байт) – 32;
- число ітерацій в одному циклі – всього 64;
- швидкість при стандартних умовах (MiB/s) – близько 140.

Робота алгоритму SHA-256 базується на принципі процедури Damgard-Merkle, згідно з яким вихідна інформація розбивається на частини однакового розміру, кожна з яких піддається обробці односторонньою функцією стиснення. Після такої операції довжина вхідного повідомлення зменшується [9].

Сформований хеш-функцією код має фіксовану довжину, незалежно від розміру вхідної інформації. Розмір отриманих образів варіюється в діапазоні від 30 до 512 біт.

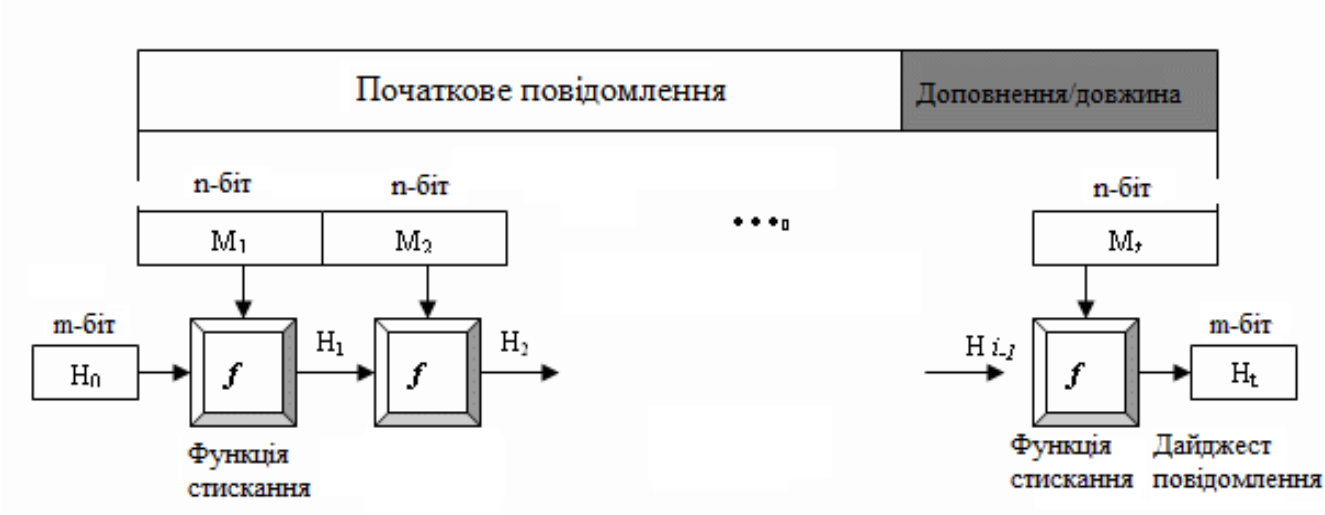


Рис. 1.6. Хешування за допомогою метода Меркле – Дамгарда

Як працює SHA-256: оригінальний текст після вставки розбивається на блоки, кожен блок розділяється на 16 слів. Алгоритм передає кожен блок повідомлень через 64 ітераційний цикл. У кожній ітерації перетворюється 2 слова, функція перетворення встановлюється з іншими словами. Сума результатів обробки кожного блоку додається, що є значенням хеш-функції. Паралельна обробка блоків неможлива, оскільки ініціалізація для внутрішнього стану є результатом попередньої обробки блоків.

Надійна хеш-функція володіє двома якостями:

– швидко обчислюється і дуже довго розшифровується без наявності ключів (дана властивість залежить від потужностей комп’ютера);
– мінімізує ризик появи колізій (дана властивість залежить від даних).

Колізії – це однакові образи, що виникають через обробку однакових вхідних блоків даних. Для боротьби з колізіями розроблені спеціальні методи. Наприклад, при роботі з хеш-таблицями застосовуються методи відкритої адресації або прямого зв'язування. Якщо необхідно захистити від фальсифікації паролі і електронні підписи, застосовується метод “криптографічної солі”, коли в створенні хешу бере участь вільно змінна послідовність бітів.

Важливо, щоб вміст образу змінювалося слідом за оригіналом. Це є першою і головною вимогою до хеш-функцій будь-якого роду. Якщо зміни не будуть відслідковуватися, то образ перестане відповідати оригіналу і робота з даними виявиться неможливою.

Друга вимога – унікальність кожного образу. Звичайно, існує ймовірність того, що образи двох файлів або баз даних співпадають, але вона надзвичайно мала. Чим вище надійність алгоритму, тим ця вірогідність менше. Хеш-функції сімейства SHA-2 досить захищені від колізій, тому збої в роботі систем практично виключені.

Третя вимога – однонаправленість функції. Алгоритм працює тільки в одну сторону, тобто стискає, перемішує і розсіює інформацію, але відновити її на підставі отриманого результату він не здатний без наявності ключів. Це служить додатковим захистом при шифруванні даних.

Четверта вимога – неможливість підробки: необхідно, щоб підбір повідомлення з правильним значенням способу володів високою складністю. Якщо значення одного з повідомлень все ж стане достовірно відомим, то підбір інших повідомлень повинен бути максимально складний.

Тобто, хешування - це процес перетворення рядка вхідних даних випадкової довжини у бітовий потік фіксованої довжини (вихідний). Наприклад, хеш-функція може приймати рядок з будь-якою кількістю знаків, а на виході отримувати рядок зі строго певним числом символів (дайджест).

Окремою категорією хеш-функцій є криптографічні хеш-функції. Вони мають значно суворіші вимоги, ніж функції, які зазвичай використовуються в хеш-таблицях. Тому їх використовують у більш «серйозних» ситуаціях, наприклад, при збереженні паролів. Криптографічні хеш- функції виробляються і ретельно перевіряються дослідниками по всьому світу.

Гарна хеш-функція забезпечує захист від колізій (неможливо отримати два однакових хеш значення при різних початкових даних) і володіє так званим ефектом лавини, коли найменша зміна вхідних даних значно перетворює вихідне значення.

Хеш-функції в блокчейн гарантують незворотність всього ланцюжка транзакцій. Кожен новий блок транзакцій посилається на хеш попереднього блоку в реєстрі. Хеш самого блоку залежить від всіх транзакцій в блоці, але замість того, щоб послідовно передавати транзакції хеш-функції, вони збираються в одне хеш-значення за допомогою двійкового дерева з хешами (дерево Меркле). Таким чином, хеші використовуються як заміна вказівниками в звичайних структурах даних: пов'язаних списках і бінарних деревах. Властивість незмінності хешу одного блоку гарантує незмінність всього блокчейн.

Отже, якщо хтось мав отримати доступ до певних даних у блоці, замість того, щоб проходити по них лінійно, він може переходити за допомогою хешів у дереві Меркле, щоб дістатися до даних. У випадку, якщо транзакції зберігались лінійно, пройти всі транзакції щоб знайти конкретну, буде дуже громіздко.

Завдяки використанню хешів загальний стан блокчейна - всі транзакції на даний момент та їх порядок - можуть бути виражені одним числом: хешем найновішого блоку. Тому право власності на хеш блоку гарантує незмінність усього блокчейну (див. рис. 1.7).

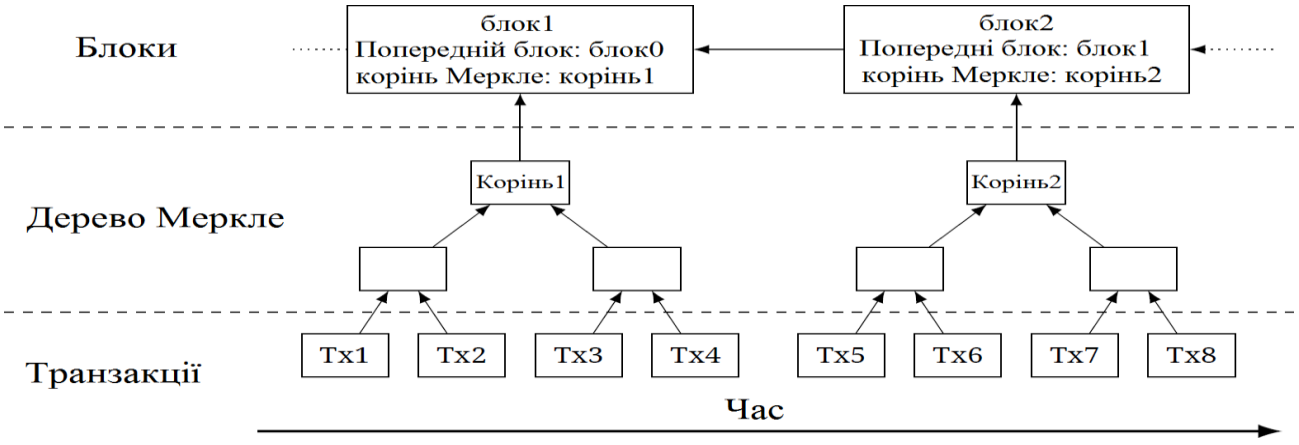


Рис. 1.7. Графічне представлення хешування блоків транзакцій

Висновки до Розділу 1

В першому розділі було розглянуто поняття інформаційної безпеки, яке є багатоаспектним та багатовимірним, наведено визначення інформації та її властивості, розглянуто поняття асиметричної криптографії, на якій базується blockchain. Вказано на особливості застосування цифрового підпису, який забезпечує цілісність даних та наведено особливості хешування, яке забезпечує незворотність всього ланцюжка транзакцій, що є основою блокчейн.

На сьогодні для запобігання та уникнення шахрайських дій з інформаційними активами необхідні нові та надійні рішення. Блокчейн може дати державним органам нові інструменти, які скоротять обсяги шахрайства, фальсифікації та число помилок. Необхідною умовою правильного застосування СЕД необхідна правильна постановка принципів роботи технології блокчейн.

РОЗДІЛ 2. АНАЛІЗ ОСОБЛИВОСТЕЙ ТЕХНОЛОГІЇ БЛОКЧЕЙН

В даному розділі основну увагу буде звернено на основні поняття технології блокчейн, загальний принцип роботи та класифікацію. Описано основні структурні блоки та алгоритми досягнення консенсусу. Дані відомості необхідні для подальшого вивчення теми дипломної роботи та практичної реалізації.

2.1 Загальні відомості про блокчейн та класифікація

Блокчейн (block - блок, chain - ланцюг) - розподілена база даних, яка постійно зростає і веде список записів, що називаються блоками. База захищена від підробки та перероблення. Кожен блок містить мітку часу та посилання на попередній блок хеш дерева [10].

Технологія блокчейн стала справді геніальним творінням розуму людини або групи людей, що працюють під псевдонімом Сатоші Накамото. З моменту винайдення та формулювання принципів роботи мережі ця технологія зазнала чималих змін та популярності у світі. Blockchain – це принципово нова надійна технологія зберігання записів, яка може кардинально змінити підхід до формування і зберігання баз даних. Вона дає можливість поширювати інформацію мережею без її копіювання між учасниками мережі – так блокчейн створив нову основу для нового типу всесвітнього інтернету. Оригінальна розробка технології була спрямована на винайдення нового слова у сфері цифрових валют – криптовалют, таких як Bitcoin, ETH (Ethereum), та інших. Але, з часом, спеціалісти в технічній сфері почали винаходити нові варіації і потенціали такого методу. Як приклад, система документообігу на блокчейн.

Блокчейн – це механізм, що забезпечує високий ступінь обліку та ідентифікації інформації, дає змогу поширювати цю інформацію між

користувачами мережі, одночасно працювати з нею декільком користувачам, фіксуючи час кожної транзакції.

Блокчейн це незламний цифровий кластер для запису змін в мережі, що може бути запрограмовано не тільки на запис фінансових транзакцій, але й будь-яких інших існуючих значень – будь-якої інформації у світі. Технологія дійсно здатна захистити дані, з якими доводиться працювати, при цьому зробивши їх більш доступними й прозорими. До того ж, блокчейн може помітно знизити витрати та мінімізувати час, необхідний для вирішення виникаючих проблем і усунення помилок.

Блокчейн - розподілений обліковий запис, який надає спільноті можливість зберігати та обмінюватися інформацією в Інтернеті. У цій системі кожен учасник зберігає свою власну копію інформації, і всі члени повинні підтвердити будь-які оновлення колективно [11]. Це можуть бути операції, контракти, активи, посвідчення особи або будь-що інше, що можна ідентифікувати цифровим способом. Записи постійні, прозорі та доступні для пошуку, що дозволяє користувачам системи переглядати всю історію транзакцій. Кожне оновлення – це новий блок, який додається до кінця кожного ланцюжка. Шифрування за допомогою блокчейну замінює сторонніх агентів, тим самим підвищуючи довіру та цілісність системи.

Блокчейн – вибудуваний за певними правилами безперервний послідовний ланцюжок (зв’язний список) блоків, які містять інформацію. Дана технологія може значно спростити відстеження підозрілих транзакцій і в цілому підвищити прозорість угод. По суті це технологія розподіленого підтвердження транзакцій, яка являє собою величезну розподілену базу даних. При цьому перевіркою достовірності транзакцій займаються самі учасники, вони ж підтверджують їх справжність та формують блоки записів. Тобто це децентралізована база даних, де всі блоки пов'язані між собою за допомогою засобів криптографії.

Саме децентралізація є однією з найважливіших концепцій блокчейн. Жоден комп’ютер або організація не може бути власником мережі.

Децентралізована база даних – така база даних, інформація в якій зберігається на різних незалежних серверах, які не пов’язані між собою єдиним власником або місцем розташування. Децентралізація блокчейн значно знижує ймовірність фальсифікації баз даних. Спосіб, за допомогою якого хакерам зазвичай вдається отримати інформацію, полягає в атаці того місця, де кластеризовані всі дані – мейнфрейма. В блокчейн це практично неможливо. Тому що вся інформація зберігається через мережу блокчейн, а тому для хакерів не існує тієї артерії даних, яку необхідно атакувати. Замість цього їм буде потрібно пошкодити одні й ті ж дані в усіх блоках. Кожна зміна в блокчейн стає помітною всім учасникам, причому вона має бути затверджена їх більшістю, то така атака зажадала б божевільного обсягу комп’ютерних потужностей, що миттєво зупиняє практично будь-якого кіберзлочинця від такого подвигу.

Основне завдання технології блокчейн – довірча передача власності на цифрові активи в не довірчий мережі без посередників. Основою технології блокчейн є розподілене зберігання інформації. Це дозволяє зберігати важливу інформацію на багатьох серверах (усіх членів мережі) одночасно, а також зберігати їх відкрито та надійно. Наприклад, ця технологія може зберігати як історію банківських операцій клієнтів, так і базу даних транзакцій, результати голосування, документи, відбитки пальців або історії хвороби. Оригінальні записи можна негайно відновити із сусідніх джерел, оскільки вони зберігаються одночасно у багатьох місцях і не можуть бути сфальсифіковані без єдиного власника, інформацію не можна викрасти.

Основна суть використання технології блокчейн полягає в тому, щоб дозволити людям, які не довіряють один одному, ділитися цінними даними безпечним та захищеним від фальсифікації способом.

Блокчейн відрізняється незмінністю даних, що зберігаються, яка досягається саме за рахунок прийомів криптографії, а не внаслідок довіри до будь-кого. Два найпростіших криптографічних алгоритми, що використовуються в блокчейн, – це хеш-функції і електронні підписи, що забезпечують цілісність транзакцій і відповідають за авторизацію.

Blockchain є ланцюжком блоків даних, які створюються і зберігаються на комп’ютерах учасників ланцюжка. Ключовим поняттям блокчейн є транзакція – єдиний спосіб змінити стан даних. Блок – це структура даних, що дозволяє зберігати список транзакцій. Вузли блокчейн мережі створюють транзакції, обмінюються ними й змінюють стан блокчейн. Найчастіше копії ланцюжків блоків зберігаються і обробляються незалежно один від одного на різних комп’ютерах. Фактично, блокчейн являє собою логіку зберігання даних, що не залежить від будь-якого центру – окремого сервера або групи серверів.

Блокчейн система працює за такими правилами: (див. рис. 2.1).

- нові транзакції розсилаються всім вузлам;
- об'єднує кожен вузол у блок і намагається знайти хеш блоку транзакцій;
- коли такий хеш знайдений, блок відправляється в мережу;
- вузли прийматимуть цей блок, лише якщо всі транзакції в ньому є правильними;
- вузли висловлюють свою згоду з новими даними, починаючи роботу в наступному блоці та використовуючи хеш попереднього як вихідні дані.

Однією з найбільш перспективних та багатообіцяючих областей застосування блокчейн є технологія захищеного документообігу, яка може стати набагато безпечнішою – насамперед внаслідок ведення децентралізованого реєстру документів, який неможливо змінити або підправити. Користувачі зможуть працювати з документами, маючи повну впевненість в їх авторстві і справжності.

Таким чином, вирішується безліч проблем, пов’язаних з недосконалістю бюрократичного апарату, небезпекою махінацій та підробки документів. На цей момент у світі вже існує безліч реалізацій подібної технології, наприклад, сервіс BlockSign (США), розроблений Нью-Йоркською компанією Basno, що представляє собою загальнодоступний реєстр, в якому зберігаються, підписані електронним способом, документи.

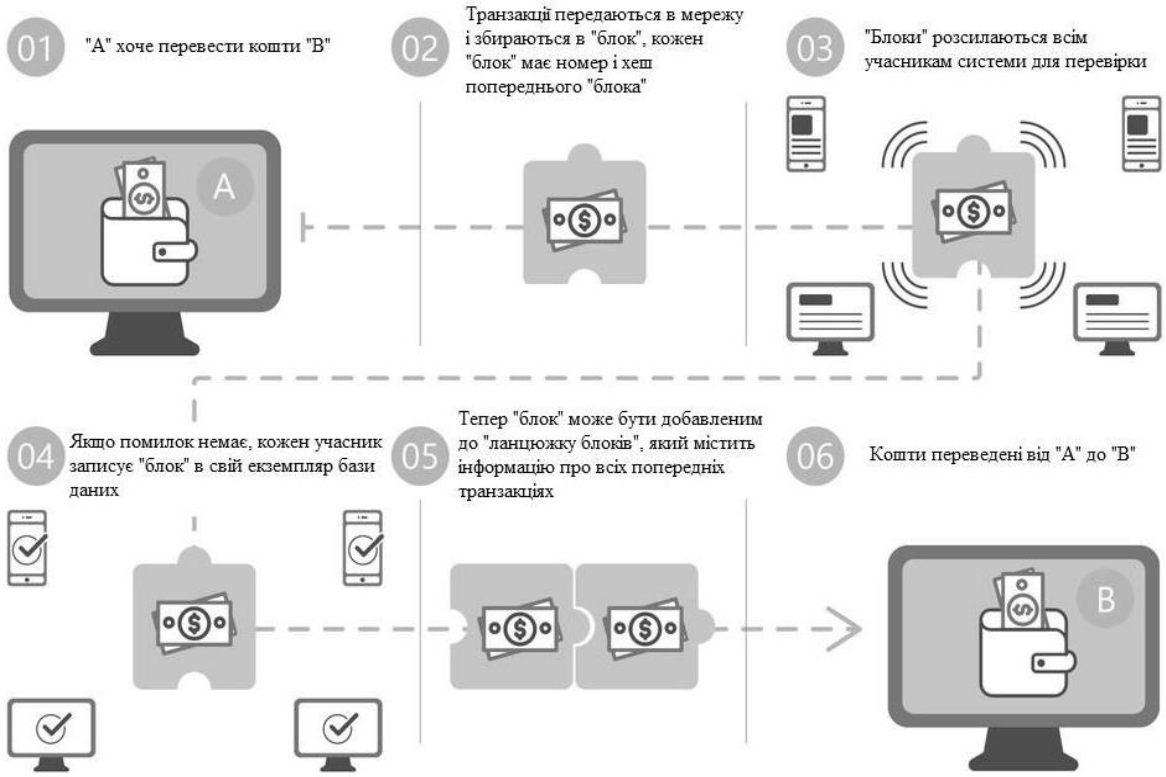


Рис. 2.1. Схема роботи системи блокчейн

Основою всіх механізмів технології блокчейну є використання наступних технологій та методів роботи та шифрування даних:

- асиметричні криптосистеми;
- хеш-функції або “хешування” даних (функції MD та SHA);
- Хеш-таблиці для запису результатів операцій хешування в блоках транзакцій;
- смарт-контракти (англ. Smart Contracts) – метод передачі даних (цифрових цінностей) від однієї особи до іншої;
- токени та реалізація механізму Proof of concept (PoC) – доказ концепції, як методу верифікації події (підтвердження угоди) в системі.

Що відноситься до класифікації блокчейн то, найбільш корисною є класифікація, дану творцем криптовалюти Ефіріум В.Бутеріним, який виділяє 3 типи блокчейн:

- публічний блокчейн з відкритим доступом (Public blockchains);

- приватний блокчейн з відкритим доступом (Consortium blockchains);
- приватний блокчейн з закритим доступом (Fully private blockchains).

Загальнодоступний блокчейн з відкритим доступом - це блокчейн, який може прочитати будь-який користувач, кожен з яких має право здійснити транзакцію. У цьому випадку транзакції захищені криптографічними механізмами перевірки, такими як підтвердження транзакції або підтвердження частки. На даний момент саме він є найпоширенішим та використовується в таких криптовалютах як bitcoin, Ефіріум, Ріплі та ін. Він насамперед добре підходить для застосування у системах документообігу, де необхідна повна незалежність від третьої особи, прозорість операцій і висока надійність системи. Також публічний блокчейн надає спосіб захисту користувачів додатків від розробників, обмежуючи можливості останніх. У додатках на публічному блокчейн розробник не може сам по собі змінювати код або дані.

Його особливостями є:

- повна відкритість блоків для всіх користувачів;
- користувачам немає необхідності довіряти один одному;
- повна децентралізація і незалежність від третіх осіб;
- захищеність мережі за рахунок криптографічних алгоритмів і застосування великої кількості комп'ютерних систем.

Приватний блокчейн з відкритим доступом – тут право завіряти транзакції, вести аудит безпеки, вносити зміни в програмне забезпечення і змінювати бази даних мають тільки привілейовані користувачі, інші отримують доступ до файлів для читання. Відповідно, така система вже не є повністю децентралізованою і залежить від певного кола осіб. Дана система може добре підійти для державних структур. Її основні властивості:

- наявність перевірених валідаторів;
- висока швидкість підтвердження транзакцій;
- неможливість проведення атаки 51% (коли в розпорядженні атакуючого знаходяться потужності більше, ніж у всій решті мережі);
- можливість видалення записів з ланцюга або їх зміни;

– більш низька вартість транзакцій за рахунок відсутності необхідності в застосуванні великих обчислювальних потужностей.

Приватний блокчейн з закритим доступом – має ті ж властивості що і попередні, але з умовою, що прості користувачі не можуть стати учасниками мережі без підтвердження валідаторів, а також не завжди мають доступ до читання файлів ланцюга. Таким чином, дана система повністю перестає бути відкритою і незалежною від третіх осіб. Отже, приватний блокчейн - це технологія, яка централізовано визначає створення блоків, і всі права на здійснення таких транзакцій належать одній організації. Користувачі можуть лише читати інформацію для контролю, але лише надійні сайти можуть керувати базами даних та іншими програмами. Властивості даної системи:

– закритість даних від непривілейованих користувачів мережі;
– повна закритість мережі від неавторизованих користувачів;
– неможливість проведення атаки 51%;
– повна залежність від кола привілейованих користувачів;
– висока надійність і стійкість мережі за умови довіри до валідатора.

Ще однією перевагою приватного блокчейну є те, що він має більше контролю над системою з боку адміністрації. Істина полягає в тому, що приватний блокчейн, наприклад, дозволяє швидко оновлювати функціональність. Тому він привабливий для організацій, що працюють з документами та системами бухгалтерського обліку та управління документами, оскільки створює контрольоване та передбачуване середовище порівняно з публічним блокчейном.

Отже, блокчейн може бути загальнодоступним або приватним. Загальнодоступний протокол - це протокол, який може прочитати будь-який користувач, кожен з яких має право діяти. Приватний передбачає, що блоки формуються централізовано, і всі права на здійснення таких операцій належать одній організації. У цьому випадку широкий загал може лише читати дані, і лише надійні вузли можуть управляти базами даних та іншими програмами. Приватний блокчейн більше схожий на класичні централізовані мережі, проте,

володіючи властивістю ведення ланцюга взаємопов’язаних блоків, може успішно застосовуватися для внутрішніх закритих приватних мереж і системах захищеного документообігу на підприємствах або державних структурах де необхідно зберігання службової інформації. Тому в даній роботі доцільним буде використання блокчейн саме цього типу.

2.2 Структурні елементи блокчейн

Блок транзакцій - електронна система управління документами та спеціальна структура для запису групи транзакцій.

Щоб транзакція вважалася дійсною, потрібно перевірити її формат та підписи, а потім групу транзакцій записати в спеціальну структуру - блок. У блоках інформацію можна швидко перевірити. Кожен блок завжди містить інформацію про попередній блок. Усі блоки можна сортувати в одному ланцюжку, що містить інформацію про всі транзакції, здійснені до цього часу в цій базі даних. Перший блок у ланцюзі - первинний блок (англ. Genesis block) - розглядається як окремий випадок, оскільки він не має основного (попереднього) блоку.

Блок містить заголовок, блок даних та список транзакцій. Заголовок містить метадані для цього блоку. Тема блоку включає хеш-значення, хеш попереднього блоку, хеші транзакцій та додаткову службову інформацію. Кожен блок охоплює два типи інформації: специфічна, яка реєструє транзакції або смарт-контракти та внутрішня, яка захищає блок і визначає, як він прив’язаний до іншого. Блоки автоматично розповсюджуються мережею, перевіряються та зв’язуються через хеш-значення.

Заголовок блоку має наступну структуру (див. рис. 2.2) [14]:

- номер блоку, також відомий як висота блоку;
- хеш-значення попереднього блоку;

- хеш-представлення даних блоку (для цього можна використовувати різні методи, наприклад, створити дерево та зберігання кореневого хешу або використання хешу всіх комбінованих даних блоку);
- позначка часу;
- розмір блоку;
- значення nonce. Коли блок генерується, він генерується випадковим чином, який потім генерує хеш заголовка блоку. Створює криптографічний хеш під час створення першого ланцюжка.

Блок даних має наступну структуру [15]:

- список транзакцій та подій книги, що входять до блоку;
- можуть бути присутні інші дані.

Блокчейн починається з блоку генезису, і нові блоки періодично додаються. Кожен блок записує виконані транзакції. Вузли працюють разом, щоб об’єднати блоки в блокчейн і сформувати книгу, яку неможливо змінити без перероблювання затвердження (PoW). Кожен вузол блокчейну має свою копію, і мережа потребує алгоритмічного оновлення, довіри та верифікації ланцюга. Завдяки прозорості технології цей процес легко контролювати та переглядати. Власні транзакції користувачів відображаються з унікальним буквено-цифровим ідентифікаційним номером, присвоєним їм.

Далі слідують всі або деякі з останніх транзакцій, які ще не записані в попередні блоки. Хеш дерева використовується для транзакцій у блоці (деревоподібне хешування). Якщо хеш заголовка менше або дорівнює певній кількості числового значення, згенерований блок буде прийнятий іншими користувачами, його значення встановлюється періодично. Результат контрольної суми незворотний (функція SHA-256), за винятком випадкового пошуку, немає алгоритму отримання бажаного результату. Якщо хеш не задовольняє умові, параметр nonce у заголовку змінюється, і хеш відображається в списку. Зазвичай для цього потрібна велика кількість перерахунків. Як тільки опція знайдена, вузол надсилає отриманий блок іншим підключеним вузлам, що керують блоком. Якщо помилки немає, блок

вважається доданим до ланцюжка і наступний блок повинен містити його хеш-значення.

Іншим важливим компонентом є ланцюговий блок. Блоки створюються одночасно багатьма майнерами. Блоки, що відповідають критеріям, надсилаються в мережу шляхом приєднання до бази даних розподіленого блоку. Зазвичай бувають випадки, коли кілька нових блоків у різних частинах розподіленої мережі називаються однаковим попереднім блоком, тобто ланцюг блоків може бути розгалуженим. Ви можете спеціально або ненавмисно обмежити повторну передачу інформації про нові блоки. У цьому випадку можливе паралельне зростання різних гілок. По мірі продовження блокової ретрансляції майнери починають розглядати основний ланцюг, беручи до уваги рівень складності хешу та довжину ланцюга. При однаковому розподілі за складністю та довжиною, три блоки вибору надаються ланцюжку, що з’явився раніше. Лише транзакції, введені у відхилену гілку, втрачають статус затвердженого. База даних розподіленого блокчейну створюється як постійно наростальний блокчейн із записами всіх транзакцій. Створення ланцюжка блоків одночасно в копії бази даних або її частини зберігається та синхронізується на декількох комп'ютерах згідно з формальними правилами.

База даних зберігає незашифровану інформацію про всі транзакції, які публічно підписані асиметричним шифруванням. Щоб уникнути багаторазового витрачання однієї і тієї ж суми, база даних використовує мітки часу, розділяючи їх на спеціальний ланцюжок блоків, кожен з яких, серед іншого, містить хеш та серійний номер попереднього блоку. Кожен новий блок підтверджує транзакції, інформація яких включає додаткове підтвердження транзакцій у всіх попередніх блоках ланцюга. Недоцільно змінювати інформацію в блоці вже в ланцюжку, оскільки в цьому випадку необхідно редагувати інформацію у всіх наступних блоках. Тому успішна атака подвійних витрат (перевитрата раніше витрачених коштів) є практично надзвичайно низькою.

Існує кілька механізмів захисту блокчейн. Перший передбачає прив’язку кожного блоку до попереднього обчислювальної складним способом скасування. Це досягається двома комбінованими методами: використання хеш-дерева, що означає, що хеш обчислюється для кожного блоку, що включає хеш-значення попереднього блоку (це робиться для кожного нового створеного блоку, за винятком першого блок генезису, який не має попередника і включення спеціального числа в кожен блок, блоку nonce. Вставка правильного nonce дозволяє розрахувати конкретне хеш-значення по всьому блоку. Коли nonce вставляється у правильному місці, обчислення хеш-функції по блоку дає конкретне хеш-значення (таке, що починається з заданої кількості нулів).

Другий механізм захисту – це вбудований механізм консенсусу однорангової мережі, за допомогою якого більшість вузлів повинні узгодити наступний блок, який розширює ланцюг. Це означає, що немає центральної точки управління, яка може бути порушена. Система Blockchain функціонує без центральної довіреної сутності, в режимі однорангового зв’язку, де всі вузли рівні. Немає довіри між вузлами – вони покладаються на механізм консенсусу для підтвердження операцій. Механізм консенсусу базується на перевірці кожним вузлом, що отримана інформація відповідає набору правил, і на перевірці nonce. Правила підтверджують, що запропонована транзакція відповідає функціональності програми, яка є специфічною для програми.

Оскільки nonce важко обчислити, заміна одного блоку іншим означала б повторне обчислення nonce всіх блоків, які згодом були з ним зв’язані. З огляду на поточний стан алгоритмів та обчислювальної потужності, як правило, це неможливо після розширення ланцюжка приблизно на шість блоків.

Отже, загальний принцип роботи блокчейн полягає в наступному. Кожному блоку присвоюється цифровий підпис, який є хеш-сумою з унікальним ідентифікатором. Завдяки математичній функції всі блоки розташовані в необхідному порядку (оскільки, як уже згадувалося раніше, термін «blockchain» буквально перекладається як блокчейн). При спробі

змінити порядок блоків система подає повідомлення про помилку, оскільки існує невідповідність структури та ідентичності.

Щоб зловмисник не зміг змінити електронний підпис і розрахувати кількість хешу, який система фактично виявить, блокчейн використовує кілька методів захисту інформації: Доказ роботи («Proof of Work») та Доказ володіння («Proof of Stake»).

2.3 Аналіз алгоритмів досягнення консенсусу

Ключовим аспектом технології блокчейн є визначення того, який користувач публікує наступний блок. Це вирішується шляхом впровадження однієї з багатьох можливих моделей консенсусу. Технологія блокчейн передбачає зберігання однієї і тієї ж бази даних на різних серверах і шифрування даних. Однак тут виникає питання про те, які окремі вузли «домовляються» між собою щодо того, яку копію слід вважати правильною, тут необхідний механізм консенсусу.

Консенсус - це механізм, який дозволяє учасникам мережі переносити кожну наступну транзакцію до загальнодоступної мережі вільно і без ризику.

Щоб додати блок до блокчейну, вузол повинен вирішити певні обчислювальні проблеми, які значно ускладнюють управління мережею вузлом. Тепер, щоб прийти до консенсусу, давайте розглянемо основні алгоритми - математичні алгоритми, що дозволяють усім користувачам системи дійти спільної згоди з певного ключового моменту.

Вперше ідея про доведення своєї легітимності і наявності певних прав за допомогою виконання трудомісткої роботи була запропонована в статті Синтії Дворко і Мни Наор “Pricing via Processing or Combatting Junk Mail” 1993 [16]. У цій статті користувачам пропонувалося обчислення трудомісткої функції для доступу до певних ресурсів, при цьому таке обчислення повинно легко перевірятися, проте бути досить трудомістким, хоча і обчислюватися в

прийнятні терміни. Дана система могла оберігати корпоративні мережі від DOS/DDOS-атак, тому що поширення величезного числа листів з одного комп'ютера ставала фізично нездійсненним. Чотирьма роками пізніше Адамом Беком був створений проект Hashcash, основною ідеєю якого було знаходження значення певного числа X, хеш якого містив би N старших біт одного значення, наприклад, нульового. У 1999 році термін Proof-of-Work вперше згадується у статті «Proofs of Work and Bread Pudding Protocols» Маркуса Якобсона та Аpi Джуелз. Цей метод став більш популярним після використання в криптовалюті біткойн: доказом роботи є пошук хешу блоку транзакцій за допомогою функції SHA-256, і до цього існують конкретні вимоги, тобто має бути заповнена певна кількість великих бітів з нулями. Згодом даний метод став поширюватися на інші криптовалюти, такі як Litecoin, Ripple, swiftcoin, etherum тощо. Найбільш поширеними є моделі "Підтвердження роботи (Prof of Work – PoW)", "Підтвердження частки (Prof of Stake - PoS)", "Делеговане підтвердження часток (Delegated proof of stake)", "Гібридна система (Prof of Work / Prof of Stake)", система "Підтвердження активністю (Prof of Activity)", "Підтвердження спалювання (Prof of burn)", "Підтвердження ємністю (Proof of Capacity)", "Підтвердження зберігання (Proof of Storage)".

Розглянемо основні алгоритми більш докладно.

a) Підтвердження роботою (Proof of Work – PoW) :

В основному це пов'язано з можливістю вузла перевірити, чи дійсно зроблені обчислення майнера (що є вузлом, який додає новий блок до блокчейну). У блокчейні вузли генерують ітерації у так званому «одноразовому» порядку. Сюди входить спроба знайти заголовок блоку, який відповідатиме цьому рівню складності (частина блокчейну, що містить посилання на попередній блок і містить остаточне значення транзакцій, включених до блоку).

Ці розрахунки можна проводити лише в Інтернеті, а складність регулюється до рівня, який є справді складним. Результати розрахунків залишаються простими для перевірки. Вузли завжди можуть бути впевнені, що

майнер знаходить правильне значення. Дана модель передбачає учасникам підтверджувати свою дію виконанням трудомісткої роботи, при цьому робота повинна вимагати більших, але прийнятних за часом обчислювальних витрат, і повинен існувати спосіб швидко перевірити виконання цієї роботи. “Суть полягає в пошуку такого значення, чий хеш (наприклад, SHA-256) починався б з деякого числа нульових бітів. Потрібно виконати обсяг роботи, який експоненціально залежить від числа нулів, але для перевірки знайденого значення досить обчислити лише один хеш”. Таким чином, підібрати таку хеш функцію для блоку даних вельми проблематично, а ось перевірити вже існуючу не складе труднощів.

Дане рішення починається з сервера міток часу. Сервер мітки часу працює, беручи хеш блоку елементів, що підлягають мітці часу, та широко публікує хеш. Відмітка часу доводить, що дані, існували на той момент, для того щоб потрапити в хеш. Кожна мітка часу включає попередню мітку часу у свій хеш, утворюючи ланцюжок, причому кожна додаткова мітка часу підсилює ті, що перед нею. Цей метод вирішує проблему прийняття рішення більшістю. Рішення більшості представлено найдовшим ланцюжком, який має найбільші зусилля, спрямовані на підтвердження роботи. Якщо більшість потужностей центрального процесора контролюється чесними вузлами, чесний ланцюг буде рости найшвидше і випереджатиме всі конкуруючі ланцюги. Щоб змінити минулий блок, зловмисникові довелося б повторити перевірку роботи блоку та всіх блоків після нього, а потім наздогнати та перевершити роботу чесних вузлів.

До плюсів даного методу можна віднести:

1) відносна стійкість системи – зловмиснику необхідно заволодіти 51% всієї потужності мережі, щоб змінити ланцюжок;
2) незалежність від третіх осіб – зовсім необов'язково довіряти будь-кому з користувачів мережі, немає необхідності в третіх особах для завірення угоди.

До мінусів відносяться:

1) марне витрачання обчислювальної потужності всіх комп'ютерів при підтвердженні роботи;
2) низькі швидкості обробки транзакцій всередині мережі;
3) можливість об'єднання майнерів в співтовариства (пули) з подальшим нарощуванням обсягів, що, в підсумку, породжує монополію на процес підтвердження роботи великими організаціями, які в підсумку можуть укласти конгломерат і захопити 51% потужності всієї мережі.

б) Підтвердження частки (Prof of Stake - PoS) [18]:

Дана модель має на увазі досягнення консенсусу свого роду голосуванням, на якому кожен як доказ приводить свої активи в якості ставки. Таким чином, якщо запропонований варіант виявиться неблагонадійним, учасник, який запропонував свій варіант, втратить всі свої активи, і шахрайство стає невигідним. Перевагою даної моделі є:

1) зниження витрат на підтримку системи – для підтвердження транзакцій ми не потребуємо складних обчислювальних процесів;
2) відсутність необхідності постійного поліпшення обчислювальної техніки і нарощуванні технічних обсягів;
3) атака на систему стає набагато дорожчою, тому що якщо зловмисник захоче купити 51% монет, курс відразу ж виросте, та й навряд чи буде сенс скуповувати більше половини всієї валюти заради махінацій в ній же самій.

Однак, дана система має ряд серйозних недоліків:

1) мотивує учасників до накопичення великої кількості коштів, що веде до децентралізації самої системи;
2) групи учасників можуть об'єднуватися в групи і створювати власні фінансові сили шляхом тиску на інших учасників мережі;
3) Можливість подвійних витрат, тобто: зловмисник може створити свій власний ланцюжок, витрачаючи неіснуючі ресурси, це буде довше, ніж легітимне, і чесні користувачі можуть це підтримати, оскільки вони не використовують реальні ресурси (атака Nothing-on-stake).

в) Делеговане підтвердження часток (Delegated proof of stake):

Модель схожа на модель підтвердження частки, однак, відмінністю є можливість володіючих активами користувачів “здавати в оренду” частину своїх коштів для проведення майнингу. Користувач, у якого багато валюти, може здати частину своїх грошей іншим, які не можуть витрачати її, але можуть використовувати її для голосування додання блоку в ланцюг. Дана поправка стимулює більшу кількість користувачів займатися майнингом, так як для цього тепер немає необхідності мати великі суми валюти.

г) Гібридна система “PoW / PoS”:

Система поєднує як створення блоків і додавання їх в ланцюжок за допомогою майнингу (PoW), так і за допомогою підтвердження своїми депозитами (PoS). Сам ланцюжок складається з обох типів блоків. Таким чином, переписати попередні блоки обманним шляхом, використовуючи вразливості Proof of Authority (PoA), стає набагато складніше, тому що існують так звані “точки контролю” у вигляді блоків PoW. Також, завдяки майнингу блоків, можливо виробляти емісію грошей за принципом стандартної PoW, а підтвердження блоків шляхом PoS можна використовувати як додаткове джерело доходів.

Однак, вразливість типу “nothing-on-stake” все ж залишається, так як є можливість знаходження блоків PoS по всій довжині ланцюжка і відповідно використання ключів витрачених грошей для підтвердження альтернативної гілки блоку зловмисника.

д) Підтвердження активністю (Prof of Activity):

Рішенням проблеми “nothing-on-stake” в гібридних системах може послужити використання для кожного блоку як PoW-майнингу, так і PoS. На даний момент ця система існує тільки теоретично, проте, має величезний потенціал використання. Вона передбачає, що учасники PoS вступають в голосування тільки після того, як майнерами проведена певна частина роботи, тобто навіть якщо який-небудь користувач буде володіти 51% монет, він не зможе одноосібно контролювати створення блоків.

е) Підтвердження спалювання (Prof of burn):

Досить рідкісна і малопоширена концепція, при якій користувачі відправляють монети на адреси, з яких неможливо зняти гроші назад, тобто свого роду “спалювання” коштів заради отримання права на довічний майнинг нових токенів. Ця модель є всього лише теоретичною і до теперішнього моменту не використовувалася в великих проектах.

ж) Підтвердження ємністю (Proof of Capacity):

Алгоритм являє собою систему, подібну підтвердження частки, яка використовує в якості показника довіри не рахунок вкладу, а кількість виділених мегабайт на комп’ютері. Алгоритм побудований таким чином, що створює на жорсткому диску великі блоки даних шляхом багаторазового хешування ключа з якимись випадковими числами, і в кінці кожного блоку створює індекс-мітку. Чим більше пам'яті виділено – тим більше міток в наявності, відповідно, більше шансів на підтвердження блоку ланцюга.

Перевагами даного алгоритму є:

– захист від бот-мереж, тому що досить легко виявити раптове переповнення жорсткого диска комп'ютера;
– відсутність необхідності витрати енергії даремно, як при алгоритмі підтвердження роботою.

Недоліки:

– можливість атаки nothing-on-stake;
– досить висока швидкість роботи.

з) підтвердження зберігання (Proof of Storage):

Алгоритм досягнення консенсусу схожий з попереднім, проте, основна ідея в тому, що виділений дисковий простір використовується всіма учасниками мережі як хмарне сховище.

и) Підтвердження компетентністю:

Алгоритм, в якому всі користувачі мають свій ранг в мережі, в залежності від якого змінюється шанс запевнення користувачем блоку. Наприклад, всі користувачі можуть ділитися на звичайних, експертів та адміністраторів. Шанс

майнингу блоку для звичайного користувача може бути дорівнює 1%, для експерта 20%, а для адміністраторів 60%.

Проаналізувавши можливі варіанти реалізації блокчейн в залежності від приватності користувачів і закритості інформації, можна побудувати графік, зображений рис.2.2.

У підсумку в лівому верхньому кутку, де ми отримуємо мінімальну довіру до валідатора і максимальну приватність користувачів, у нас розташовуються публічні мережі з відкритим доступом. Для таких мереж рекомендовано використання алгоритму досягнення консенсусу Proof of work.

У лівому нижньому кутку розташовуються приватний блокчейн відкритого типу. В таких системах низька довіра до валідатора, тому в них підвищені вимоги до алгоритмів досягнення консенсусу, прикладом таких можуть бути Proof of important, proof of activity.

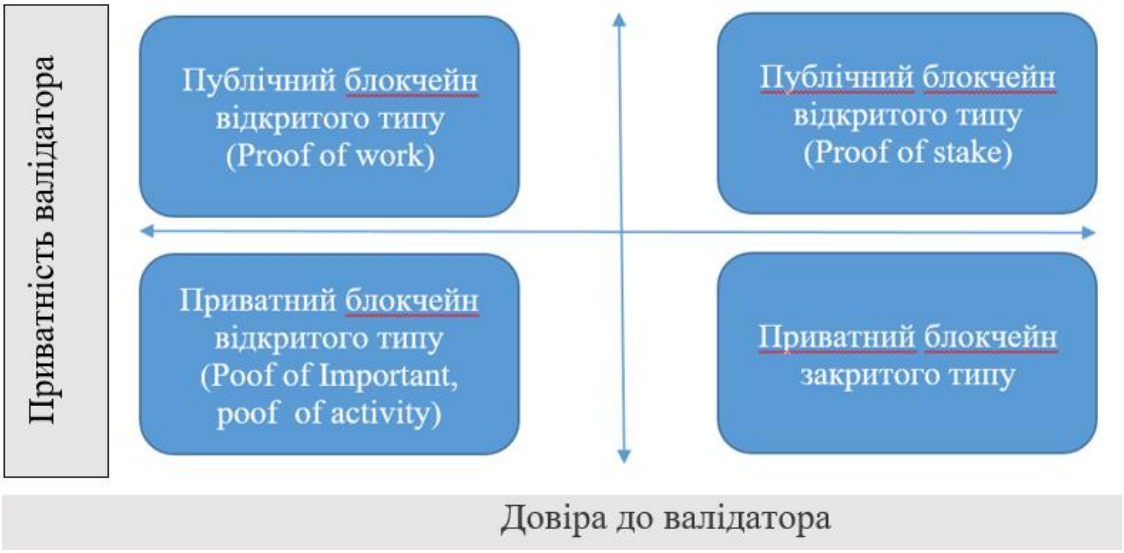


Рис. 2.2. Класифікація блокчейн по відношенню до параметрів приватності і довіри

У правому верхньому куті також максимальна приватність валідатора, однак існує довіра до валідаторів. Для таких систем підходять алгоритми Proof

of stake, де немає необхідності публічно відкривати свою особистість, однак, чесність доводиться активами, вкладеними в ставку на підтвердження блоку.

Останньою областю таблиці є правий нижній кут, який припадає на приватний блокчейн закритого типу, в яких всі центри запевнення транзакцій є відкритими, а довіра до них дуже велика. Таким чином, даний вид блокчейн швидше нагадує звичайні централізовані ієрархічні мережі, однак, може бути успішно використаний на практиці завдяки властивостям безперервного ланцюга взаємопов'язаних блоків інформації.

Якщо ж виділити конкретні етапи роботи блокчейн в дії, то алгоритм буде виглядати наступним чином:

- все починається з верифікації (з боку користувача) підтвердження на транзакцію. Це активує всю систему для користувача;
- тоді транзакція поєднується в блок із початковим станом (тобто адресацією та часом) та очікуваним кінцевим станом (якщо транзакція підтверджена);
- зазначений блок надсилається всім учасникам для перевірки початкового стану блоку;
- всі згадані учасники отримують дані про передбачувану транзакцію, не тільки підтверджуючи правильність початкового стану блоку, але й записуючи дані про блок у своєму ланцюжку. Тому немає місця, де зберігається база даних транзакцій;
- після підтвердження блоку, учасниками підтверджується весь ланцюжок. Транзакція отримує путівку в життя і здійснюється.

З цієї причини блокчейн часто є рахунком групи транзакцій, інакше відомим як блоки, які криптографічно пов'язані між собою в лінійній шкалі часу. Інші ключові особливості, пов'язані з блокчейном, - це безпека, узгодженість та конфігурація - ці атрибути залежать від архітектури блокчейну та природи консенсусного протоколу, який він використовує для побудови системи. Деякі блокчейни налаштовані для полегшення однорангових транзакцій між неієрархічними вузлами.

Висновки до Розділу 2

В другому розділі наведено поняття технології блокчейн, її особливості та класифікацію. Розглянено принцип роботи та алгоритми досягнення консенсусу.

Блокчейн складається з набору захищених інформаційних блоків, послідовно прикутих один до одного. Разом вони утворюють незмінну книгу, розподілену по вузлах, що беруть участь. Ці вузли є обчислювальними платформами, які взаємодіють із кінцевими користувачами. Метою блокчейн є обмін інформацією між усіма сторонами, які отримують до нього доступ через додаток. Доступ до цієї книги з точки зору читання та письма може бути необмеженим або обмеженим. Спільна інформація захищена від модифікації, що означає, що будь-які зміни можна буде легко та негайно виявити. З цієї причини, як тільки інформація впорядкована на блокчейн, вона вважається незмінною, оскільки вона настільки надійно захищена.

Враховуючи те, що технологія дійсно здатна захистити дані, з якими нам доводиться працювати, при цьому зробивши їх більш доступними й прозорими та може помітно знизити витрати і мінімізувати час, необхідний для розв’язання виникаючих проблем і усунення помилок. З огляду на властивості даної технології можлива реалізація системи електронного документообігу на основі блокчейн за допомогою криптографії. Тому, розгляд принципів роботи blockchain є особливо важливим в даній роботі, так як практична реалізація СЕД буде здійснена саме за допомогою цієї технології.

РОЗДІЛ 3. СИСТЕМА ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ

В даному розділі буде представлено переваги переходу на електронний документообіг принципи та властивості ефективної СЕД. Проаналізовано можливості та переваги створення СЕД на основі технології блокчейн.

3.1 Поняття та визначення електронного документообігу

У сучасному світі інформація є стратегічним національним ресурсом. Зростаюча необхідність від доступності інформації, міри розвитку та ефективності засобів її обробки та передавання призвела до появи такого поняття, як інформаційні ресурси. На відміну від інших, здатних до вимирання, вони не тільки не розмножуються, але й збільшуються під час їх використання. Система електронного документообігу є однією з форм формування, накопичення та обміну інформацією. СЕД базується на вивченні електронного документа, який характеризується низкою нових функцій та особливостей, на відміну від традиційних.

Сьогодні в усьому світі спостерігається стрімкий розвиток інформаційних і комунікаційних технологій, впровадження нових ідей, перехід до нових можливостей та засобів зв’язку. Для того, щоб надавати якісні послуги потрібно відмовлятися від застарілих методів обробки інформації та відповідати сучасним вимогам. Впровадження електронного документообігу - одна з таких важливих змін.

У відповідності з Законом України “Про електронні документи та електронний документообіг” від 22 травня 2003 року №851-IV, електронний документ - документ, в якому інформація, включаючи обов’язкові реквізити документа, записується у вигляді електронних даних. Електронний документ може бути створений, переданий, збережений та електронно перетворений у візуальну форму. Візуальною формою подання електронного документа є

відображення даних, які він містить, електронними засобами або на папері у формі, придатній для приймання його змісту людиною [19].

Електронний документообіг (обіг електронних документів) - сукупність процесів створення, обробки, відправки, передачі, прийому, зберігання, використання та знищення електронних документів за допомогою перевірки цілісності та при необхідності підтвердження отримання таких документів (за потреби) [19].

Документ є базовим елементом в системі документообігу. Ці системи забезпечують рух документів всередині підприємства, надають можливість відстежити процеси до яких належать ті чи інші документи, зберігають як самі документи так і дані про їх зміни. В кожній організації де використовується система документообігу, документ вважається основним інструментом процесу управління. Неможливо просто прийняти рішення чи наказ, дати доручення без використання документів. Під документообігом організації прийнято розуміти впорядковане переміщення документів, створених в процесі роботи відповідними посадовими особами.

Основною метою впровадження СЕД є створення ефективного середовища управління та функціонування підприємства, установи, організації, а не викорінення паперових документів. Необхідно відзначити те, що технологія СЕД в жодному разі не витісняє паперові документи. Обидві форми повинні ефективно співіснувати та давати можливість вільно застосовувати будь-яку з них, враховуючи доцільність тієї чи іншої в конкретних умовах.

Основні організаційно-правові засади електронного документообігу та використання електронних документів регламентує Закон України “Про електронні довірчі послуги” від 13 лютого 2020 року №440-IX, а також низка підзаконних нормативно-правових актів, прийнятих на їх виконання [20].

Мета цих законів - надати електронним документам юридичну силу, як і паперовим. Ці дії зокрема:

- встановлюють процедури та організації електронного документообігу;
- встановлюють організаційно-правових основ використання ЕД;

- визначають правовий статус ЕЦП;
- регулюють відносини, що виникають від час використання ЕЦП.

Згідно із Законом "Про електронні довірчі послуги", ЕЦП розміщується поверх набору електронних даних, доданих до цього кластеру або логічно поєднаних з ним. Відповідно до статті 6 ЗУ “Про електронні документи та електронний документообіг” ЕП є обов’язковим реквізитом ЕД, який використовується для ідентифікації автора та/або підписувача ЕД іншими суб’єктами електронного документообігу. Слід зазначити, що використання ЕЦП також дозволяє перевірити цілісність ЕД. Слід підкреслити, що за своїм юридичним статусом лише ЕЦП еквівалентна власноручному підпису.

Діяльність будь-якого підприємства регулюється документами. Саме вони визначають їх принципове значення як носіїв інформації. Насправді дуже великий обсяг управлінської роботи припадає саме на документи. Тому це і є ключова ланка діловодства. В останні роки сучасності постала проблема підвищення ефективності документообігу. Насамперед це пов’язано з постійним збільшенням кількості інформації, яка необхідна для прийняття важливих управлінських рішень. Тому виникає питання нерентабельності традиційних методів роботи.

На жаль, наразі існує багато проблем на шляху до успішного впровадження та використання системи електронного документообігу в Україні, які пов’язані, перш за все, із відсутністю ефективного механізму його впровадження, відповідності технічної баз сучасним потребам. Наразі існує досить багато рішень зберігання та форматування документів, але вони використовують клієнт-серверну модель. Для зберігання документів ми можемо використовувати хмарне і локальне сховище.

Найбільш доступним місцем для зберігання електронних документів є локальне сховище. При його використанні не доцільно застосовувати спеціальне програмне забезпечення. Користувачу достатньо впевнено користуватися операційними системами Windows чи будь-якою Unix подібною. За необхідності забезпечення доступу до електронних документів іншим

користувачам можна створити мережеву папку з налаштованим правами доступу, наприклад, тільки для читання. Однак це не гарантує безпеку цього файлу, тому Ви повинні самі забезпечити свою надійність. Крім того, термін придатності потрібно буде контролювати вручну, оскільки простір часто дуже обмежений. Розглядаючи локальне сховище з точки зору програмного забезпечення для зберігання електронних документів з електронним підписом, можна констатувати, що ця модель не підходить для цієї мети, оскільки файл підпису може бути видалений третьою стороною, а також оригінал.

Найбільш розповсюдженим та сучасним програмним забезпеченням є зберігання електронних документів у хмарних сховищах. Хмарне сховище - це модель для збереження електронних документів, де вони зберігаються в логічних сховищах, а фізичне сховище включає кілька серверів. Зазвичай фізичне середовище належить хостинговим компаніям, які керують цим середовищем. Хмарні сховища відповідають за зберігання та доступ до файлів та фізичного середовища. Розглядаючи модель хмарного зберігання електронних документів з електронним підписом, слід зазначити, що процес поділяється на два етапи. Перший етап - це коли користувач повинен підписати файл за допомогою сервісу для підписання електронних документів. Після отримання підпису потрібно надіслати файл у хмарне сховище разом з підписом. Слід відзначити те, що підпис ніяк не асоціюється з самим файлом, що є недоліком, так як при необхідності отримати підпис файлу, користувач власноруч повинен відшукати його у виділеному йому сховищі.

Найпоширенішими системами на їх основі є:

Google Drive - це компанія, що розміщує файли, створена та підтримується Google. Його функції включають в себе Інтернет-зберігання файлів, їх спільне використання та спільне редагування. Однак для цієї системи неможливо синхронізувати папки, крім каталогу Google Drive, і для Linux не існує програмного забезпечення.

Microsoft OneDrive – це файловий хостинг, що надається компанією Майкрософт як частина набору онлайн-послуг. Він дозволяє користувачам

зберігати файли, а також інші особисті дані, такі як налаштування Windows або ключі відновлення BitLocker у хмарі. Однак ця система не має можливості переглядати історію змін файлів, неможливо синхронізувати папки за межами каталогу SkyDrive, і для Linux не існує програмного забезпечення.

Dropbox дозволяє користувачам створювати приватну папку на своєму комп'ютері, яку Dropbox синхронізує, щоб мати однаковий вміст, незалежно від того, який пристрій відображається. До файлів у цій папці також можна отримати доступ через веб-сайт Dropbox та мобільні програми.

Confluence – дана система є відкритою та є єдиним місцем для пошуку інформації, обміну та спільної роботи всередині проекту або компанії. Це обговорення ідей, визначення завдань, звіти про виконану роботу за проектами, публікування і обмін документами. До основних можливостей належать: централізація, робота з файлами та документами, пошук інформації, завдання, обговорення, ідеї та рішення. Головним недоліком є відсутність захисту від несанкціонованого доступу та відсутність можливості накладання електронного підпису. Але найбільшою проблемою цих рішень є відсутність зберігання інформації без можливості модифікації.

Багато підприємств сучасного світу намагаються створити власну систему електронного документообігу, тим самим не застосовуючи хмарні загальнодоступні технології, оскільки витік інформації є цілком можливим явищем.

Застосування існуючих методів та програмного забезпечення в галузі документообігу дозволяє створювати високопотужні та повністю функціональні рішення. Однак не всі рішення забезпечують стовідсотковий захист даних перед змінами, і якщо створюються відкриті бази документів, вони можуть бути модифіковані під впливом особи, яка керує всіма даними, що є недоліком.

Для вирішення поставленої проблеми у сфері документообігу пропонується використання технології блокчейн у децентралізованій системі.

3.2 Принципи ефективної системи електронного документообігу

Ефективна робота працівників - це завжди гарантія якісного обслуговування. У наш час звичні методи обробки інформації застаріли, тому необхідно скоротити витрати часу та ресурсів на виконання завдань із використанням інформаційних технологій. Електронний документообіг - це сучасний та технологічний підхід до підвищення якості та швидкості роботи державних установ та організацій. Автоматичне дублювання документів, моніторинг історії документів в організації, контроль конфіденційності даних, значно зменшує втрати часу співробітників. Автоматична система підвищує якість роботи клерків через контроль кожного виконання етапів роботи, дозволяючи розширити можливості в управлінні персоналом, щоб дати більш точну оцінку з точки зору виконання доручень [21].

СЕД часто має інтеграцію з багатофункціональними сховищами даних. Це дозволяє впорядковувати та комбінувати дані, дозволяючи легко і швидко створювати та аналізувати звіти. Також можна знайти закономірності в збереженій інформації, щоб ви могли приймати обґрунтовані та ефективні рішення, використовуючи аналіз даних. Всі ці функції доступні лише в електронних системах управління документами. Такі системи значно спрощують процеси управління даними порівняно з паперовими документами. Це основні рішення, що забезпечують автоматизацію і централізацію обміну даних та агрегацію даних лише з потрібних джерел. Системи електронного документообігу сприяють покращенню організаційної культури, роблячи працю легшою, продуктивнішою та більш значущою. Вони дають можливість якісно вирішувати різноманітні проблеми спільними зусиллями та дозволяють прискорити швидкість роботи.

Основним завданням документообігу та електронних документів є підвищення ефективності та якості підприємств шляхом забезпечення прозорої системи переміщення документів та контролю за їх виконанням. Для того, щоб

така система була ефективною, її потрібно впроваджувати у всю роботу, пов’язану з організацією та зберіганням інформації. Іншими словами, необхідно забезпечити єдине інформаційне середовище підприємства.

До основних переваг впровадження СЕД на підприємстві є:

- економія коштів. Впровадження системи дозволяє економити гроші, призначені для копіювання та тиражування обладнання та витратних матеріалів, а також зменшення витрат на зберігання паперових документів, звільнення фізичного простору, особливо для документів тимчасового зберігання;
- економити робочий час. Процедури складання та затвердження документів, передачі, реєстрації, копіювання, пошуку за допомогою СЕД роблять це набагато швидшим;
- встановлення єдиної інформаційної області в масштабі підприємства та впровадження всіх процесів у системі;
- швидке та прозоре проходження документів, тобто забезпечується чіткість інформації;
- оптимізація процесів управління документами, вдосконалення контролю за інформаційними потоками;
- високошвидкісний пошук документів;
- запобігати значній втраті інформації через недбалість персоналу;
- запобігання несанкціонованому доступу до інформації за рахунок підвищення рівня захисту інформації за допомогою механізмів ЕЦП;
- полегшити оформлення документів в організації;
- централізоване, структуроване та систематичне зберігання документів в електронних архівах.

Основними особливостями електронного документообігу є:

- єдиний запис, що дозволяє ідентифікувати документ;
- можливість паралельної обробки для зменшення часу, відведеного на переміщення документів, що покращує ефективність виконання;

- безперервність документа, що дозволяє визначити, хто відповідає за завдання, пов'язане з документом, незалежно від стадії життєвого циклу документа;
- уникнення можливості копіювання документів за допомогою єдиної бази даних;
- ефективна організація функції електронного пошуку документів, що дозволяє знайти документ із найменшою кількістю ключових слів;
- як частина інформаційної підтримки, управління документами включає моніторинг інформаційних потоків у бізнесі, обробку, отримання та використання даних.

До основних вимог ефективної СЕД належить:

- забезпечення надійного зберігання ЕД;
- забезпечення таких можливості роботи з документом як: створення, редагування, публікація, забезпечення цілісності та доступності, зберігання;
- підтримувати обробку різних типів документів та даних пов’язаних з ними;
- забезпечувати можливість категоризації для пошуку документів;
- накладання цифрового підпису на документ;
- забезпечувати розподіл доступу за ролями в системі на основі структури організації;
- можливість контролю та управління системою для ведення історії подій;
- підтримка віддаленого доступу до системи.

Тому ідея електронного документообігу створює вже добрі технологічні передумови для поліпшення якості управління та сприяє формуванню цілісної системи електронного документообігу. Основною проблемою традиційної технології документообігу є практична неможливість відстеження руху документів у центральній організації, перевага електронного документообігу над традиційною є незаперечною. Електронний обіг документів ефективніший за папір, оскільки його легше оптимізувати. Насамперед система здатна

забезпечити цілісність та доступність, що є великою перевагою при впровадженні СЕД.

Здатність автоматизувати процеси за допомогою електронних документів, їх координація, використання ЕЦП при підписанні документів є важливими перевагами електронного документообігу. Іншим важливим критерієм у виборі системи автоматизованого документообігу є використання спільного сховища для документі. Це підвищить ефективність підготовки та обробки документів, з'являються можливості використовувати роботу інших співробітників, аналітичні матеріали, звіти, наукові знання.

Перехід на електронний документообіг значно скорочує часові витрати на виконання дій, не пов’язаних з обслуговуванням громадян: автоматичну реєстрацію документів, слідкуванню за їх переміщенням, контролюванню виконання документів тощо. Впровадження систем електронного документообігу допоможе поліпшити прийняття управлінських рішень, прискорити взаємодію, зробити обробку інформації більш якісною та швидкою.

Система електронного документообігу створена щоб забезпечити виконання таких процесів як: створення та управління великих обсягів документів, поширення їх мережею, управління доступом до даних, контроль використання документів. Дана система спроможна підтримувати безліч типів файлів: текстові документи, електронні таблиці, аудіо-, відео-файли, веб-документи та ін.

Для реалізації системи електронного документообігу, яка повинна забезпечувати надійне зберігання файлів без можливості внесення змін в систему, видалення чи модифікації файлів пропонується проаналізувати основні поняття захисту інформації технології блокчейн, звернути увагу на значимість інформаційної безпеки для ефективної системи електронного документообігу.

3.3 Аналіз можливостей технології блокчейн для побудови системи електронного документообігу

В епоху швидкого росту інформаційного середовища та збільшенням обсягів інформації з'являється необхідність цифрової трансформації. Для таких цифрових перетворень необхідно використовувати новітні технології, зокрема технологію blockchain. Задача створення ефективної системи електронного документообігу важлива не лише для успіху внутрішньої діяльності, але і для зовнішніх контактів, коли питання довіри між структурами виходить на перший план. Саме тому при створенні сучасних електронних систем документообігу доцільно використовувати нові інноваційні інструменти, зокрема, і блокчейн.

Однією з найбільш перспективних областей застосування блокчейн є система електронного документообігу, яка може стати набагато безпечнішою. Перш за все це відбувається за рахунок ведення децентралізованого реєстру документів, який неможливо змінити або поправити. Користувачі мають змогу працювати з документами маючи повну впевненість в їх авторстві та справжності. Як наслідок, зникають проблеми пов'язані з підробкою документів.

Управління цифровими документами та записами підвищує продуктивність та організаційну ефективність. Найбільш часто використовуваними функціями управління документами є відстеження версій, перевірка змін, структури та змісту документів, а також спрощений та надійний обмін документами. Блокчейн може бути корисним у кількох аспектах процесів управління документами. Наприклад, щоразу, коли створюється нова версія документа, вона може бути зареєстрована на блокчейн. Оскільки кожен новий блок у блокчейн має позначку часу, стає зрозуміло, яка версія документа була створена, а також внесені зміни, структура та вміст документа можуть бути відстежені та перевірені при необхідності. Реєстрація в блокчейн може надати необхідний доказ того, що документ не був підроблений. З іншого боку,

документи часто підписуються цифровим підписом. Після того, як вони стануть записами, їх більше не слід змінювати, а в процесі управління документами та архівування їх автентичність, цілісність, надійність та зручність користування повинні залишатися недоторканими, тоді як деякі з них також повинні зберігати характеристики відмови, безпеки та конфіденційності.

Створення сучасних електронних систем управління документами базується на таких принципах:

- разова реєстрація документів;
- паралельне виконання різних операцій з метою скорочення часу переміщення документів та підвищення ефективності виконання;
- безперервність руху документів;
- централізоване зберігання документів що виключає з документації єдину базу даних документації для дублювання;
- надійно структурована система пошуку документів.

Впровадження принципів електронного документообігу за допомогою сучасного апаратного та програмного забезпечення на базі найсучасніших інформаційних технологій створить єдиний інформаційний простір електронного документообігу шляхом інтеграції всіх систем передачі та прийому в інформаційний вузол ЕД. Інтеграція повинна працювати без втрати якості роботи. Однією з основ цієї інтеграції є система надійного зберігання та передачі даних та система взаємодії записів електронних документів.

Для вирішення поставленої проблеми у сфері документообігу пропонується використання технології блокчейн у децентралізованій системі.

Намагаючись прискорити процес ефективного впровадження системи електронного документообігу, використовують нові різноманітні рішення цілком неапробованих технологій, зокрема технологію блокчейн. При цьому вона має ряд переваг – прискорення переходу до взаємодії між працівниками структури, прозорість операцій та взаємоконтроль над їх здійсненням. Впровадження цієї технології дозволить забезпечити надійну синхронізацію даних, що унеможливить їх підміну в результаті зовнішнього втручання,

гарантує прозорість, а також дасть можливість здійснювати суспільний контроль за системою.

Будь-яка система електронного документообігу повинна працювати таким чином, щоб користувачі в будь-який момент могли отримати доступ до бази даних та перевірити її. При цьому повинна забезпечуватись незмінність документів та їх захист.

Переваги впровадження СЕД на блокчейн:

- доступність – СЕД дозволяє користувачам отримати доступ до системи в будь-який час та з будь-якого місця за умови, що вони будуть підключені до інтернету. Це дозволить забезпечити більш швидкий доступ до документів та зручність ознайомлення з ними;
- ефективність витрат – дозволяє обмежити витрати на виробництво та зменшує обсяг друкованих паперів;
- безпека та конфіденційність – правильно розроблена система має гарантії захисту інформації;
- прозорість – наочний вигляд всіх документів системи.

У технології блокчейну безпека забезпечується через децентралізований сервер, мітки часу та однорангові мережеві з'єднання. В результаті створюється автономно керована база даних без єдиного центру. Це робить його дуже зручним блокчейном для управління ідентифікацією та перевірки походження після обробки подій та транзакцій даних (наприклад, під час введення нових документів).

Децентралізовані системи характеризуються дуже високою стійкістю до зломів учасників або атакам на мережу в цілому. Немає якогось одного загального “командного центру”, зламавши який вийде знищити всі дані про угоду та її учасників або підмінити їх.

Безпека системи блокчейн полягає в тому, що записи зберігаються в зашифрованому вигляді одночасно у всіх учасників системи і автоматично оновлюються при кожній внесеній зміні. Користувачі виступають в якості колективного нотаріуса, який підтверджує істинність інформації в базі даних і

забезпечує захист від маніпуляцій і зловживань. Якщо окремий комп'ютер піддасться хакерській атаці або один з учасників мережі спробує зшахраювати, всі зможуть відслідкувати це. У проектованої СЕД на платформі блокчейн всі учасники безсумнівно повинні обмінюватися і спільно працювати з документами, фіксуючи і перевіряючи всі операції (транзакції), групувати їх у блоки і розподілено зберігати. Поняття транзакції у розрізі блокчейн технології розуміється як запит до системи для збереження одиниці інформації, тобто у конкретному випадку – файлу.

Що стосується системи електронного документообігу, то технічно ця платформа дозволяє користувачам дійти згоди про що завгодно без посередників, що забезпечує основу для децентралізованих форм управління і соціальних контрактів, заснованих на принципі консенсусу, і дозволяє підтримувати баланс в інтересах суспільства. Легкий доступ до всіх документів та забезпечення їх незмінності в приватному блокчейн є ключовою перевагою для створення системи електронного документообігу на blockchain.

Блокчейн дозволяє нівелювати вплив егоїстичних чинників, які ведуть до створення шахрайських і корупційних схем, які підривають суспільний інтерес і державний суверенітет. У той же час з'являється стимул для учасників працювати чесно, так як правила застосовуються до всіх в рівній мірі. Так виникає нова форма соціальної відповідальності. Використання даної технології дозволить забезпечити абсолютно новий підхід до обробки інформації. Така система на сучасному підприємстві є нагальною потребою. При мінімальних витратах на технічне оснащення і програмне забезпечення впровадження системи СЕД дозволяє істотно підвищити якість і продуктивність роботи з різноманітною документацією та зробити обробку інформації більш зручною для ознайомлення.

Таким чином, блокчейн володіє наступними важливими характеристиками:

– автономний – тобто немає ніякої організації, центру або агентства, які його адмініструють і мають "ключ" до виправлення даних;

- це працює цілодобово - цілий рік - оскільки вміст баз даних постійно копіюється на велику кількість комп'ютерів, навіть якщо 99% з них у певний момент перебувають у автономному режимі, записи залишатимуться в решті та оновлюватимуться як якомога швидше. Єдиним можливим способом зупинки його роботи представляється тільки відключення інтернету і електрики;
- він є безпечним – код, який використовується для блокчейн, відкритий для доопрацювання (open-source). Є можливість перевірити, чи було змінено його зміст (так званий криптографічний аудит);
- він відкритий для розробки продуктів (програм, сервісів) на його підставі і не належить будь-якої корпорації, не охороняється авторським правом або правом на інтелектуальну власність. Кожен може при бажанні провести аналіз і аудит коду.

В контексті управління документами та записами, а також з урахуванням усіх характеристик блокчейн, а також основних технологій та концепцій, можна зробити висновок, що блокчейн може бути використаний для:

- підтвердити цілісність запису;
- переконатись, що запис існував або був створений у певний момент часу (встановлення мітки часу);
- підтвердити послідовність записів;
- неможливість відмови від запису;
- накладання цифрового підпису.

У проектованої СЕД на платформі блокчейн всі учасники безсумнівно зможуть обмінюватися і спільно працювати з документами, фіксуючи і перевіряючи всі операції (транзакції), групувати їх у блоки і розподілено зберігати. Кожен авторизований учасник інформаційного обміну в такій СЕД матиме особистий кабінет, де буде зберігатися історія його операцій. Система має забезпечувати цілісність та доступність.

Отже, СЕД на блокчейн повинна мати наступні вимоги до розробки:

- кожен авторизований користувач повинен мати доступ до системи;

- несанкціонований доступ до документів повинен бути неможливий;
- змінювати, підроблювати чи видаляти документи з реєстру повинно бути неможливо;
- кожен документ повинен мати автора та бути підписаним.

Висновки до Розділу 3

В даному розділі розглянуто переваги переходу на електронний документообіг принципи та властивості ефективної СЕД, можливості створення СЕД на основі технології блокчейн. Вказано на основні переваги побудови СЕД на блокчейн.

Застосування блокчейн технології знизить витрати на підготовку звітності та забезпечить прозорість всіх операцій. Технологія надає інструменти для зниження обсягів шахрайства та виникнення помилок пов’язаних з документообігом. Даний підхід до збереження електронних документів дозволяє отримати доступ до документів будь-який час та підвищує надійність збереження документів у тому вигляді, в якому вони потрапили до сховища. Тому, розгляд можливостей технології blockchain для побудови системи електронного документообігу є особливо важливим в даній роботі, так як програмна реалізація системи буде здійснена саме на основі цієї технології.

РОЗДІЛ 4. АЛГОРИТМ РЕАЛІЗАЦІЇ СИСТЕМИ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ НА ОСНОВІ ТЕХНОЛОГІЇ БЛОКЧЕЙН

В четвертому розділі створено модель програмного забезпечення, описано технології та компоненти, які використовуються для реалізації системи, реалізовано систему електронного документообігу на основі технології blockchain.

4.1 Модель програмного забезпечення для зберігання електронних документів з цифровим підписом

Алгоритм реалізації системи електронного документообігу за допомогою криптографії та технології blockchain будується на формуванні блоків з даних, що вводяться. В кожному блоці міститься службова інформація, що необхідна для забезпечення цілісності і незмінності введених даних, а також для роботи ланцюжка блоків. Даний ланцюжок є однозв’язним списком.

Блок даних містить наступні поля (див.рис.4.1):

- index. Вказує на номер блоку в ланцюжку;
- timestamp. Мітка часу, що вказує на час створення транзакції;
- username. Ім’я користувача, що здійснив транзакцію;
- file name. Назва файлу, що доданий до системи;
- data. Значення файлу в шістнадцятковій системі;
- hash. Унікальний значення, створене на основі даних що зберігаються, який генерується алгоритмом SHA256;
- previous Hash. Покажчик попереднього хеш-блоку, що необхідний для зв’язування блоків в єдиний ланцюг.

При створенні блоку даних виконується хешування всіх збережених даних блоку з використанням алгоритму SHA256, після чого результат записується в поле Hash. Даний підхід дозволяє гарантувати незмінність даних,

тому що при зміні хоча б на один символ одного з полів блоку хеш функція поверне зовсім інший результат. Дане втручання та некоректність ланцюжка система зможе легко визначити, виконавши повторне хешування блоку та порівнявши зі збереженим хешем.

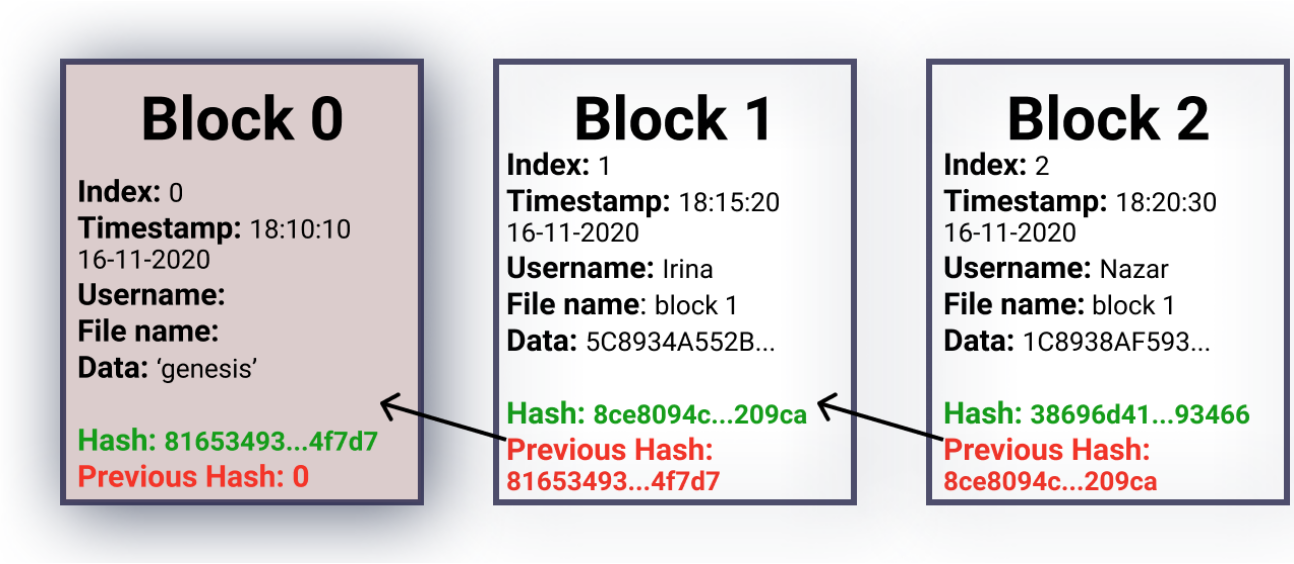


Рис. 4.1. Ланцюжок блоків blockchain

Для забезпечення підтвердження авторства в системі можливе накладання цифрового підпису. Накладання цифрового підпису забезпечує насамперед автентифікацію користувача в системі. Коли користувач підписує документ, він зберігається в блокчейн для підтвердження справжності. Завдяки блокчейну як компоненту серверної бази, можна забезпечити незмінність та відстеження записів, що зберігаються. Схема накладання ЦП здійснена за допомогою алгоритму з відкритим ключем RSA [22]. Даний алгоритм дозволяє підписати та перевірити повідомлення. При реєструванні користувача в системі відбувається формування приватного ключа, який необхідний для підписання ЕД.

Схема цифрового підпису відбувається наступним чином (див. рис. 4.2):

– генеруємо пару публічний-приватний ключ;
– використовуючи приватний ключ Аліса підписує документ та надсилає до системи;

– використовуючи публічний ключ Аліси, Боб може перевірити справжність підпису.

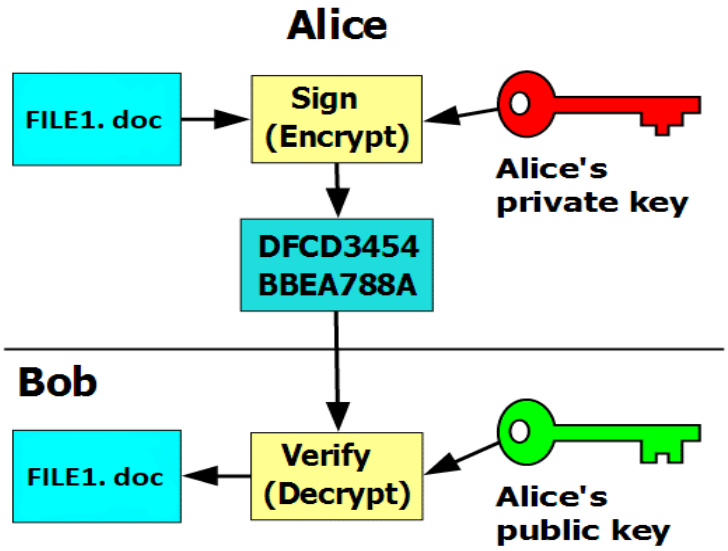


Рис. 4.2. Схема цифрового підпису

Для створення СЕД на blockchain необхідно виділити ключові етапи, результатом проходження яких є збережений ЕД у децентралізованому сховищі з накладеним ЦП .

Вхідними даними від авторизованого користувача є документ. Результатом проходження всіх етапів є підписана версія файлу збереженого до сховища. Головним процесом програмного забезпечення є збереження електронних документів. У результаті проходження всіх внутрішніх процесів збереження ЕД отримуємо неперервний ланцюжок блоків, які складають сховище ЕД з накладеним ЦП (див. рис. 4.3). Авторизація користувача відбувається після введення вірних логіну та пароля, користувач авторизується в системі і може виконувати певні дії відповідно до своїх прав в системі. Авторизований користувач має змогу завантажити документ для детального вивчення.

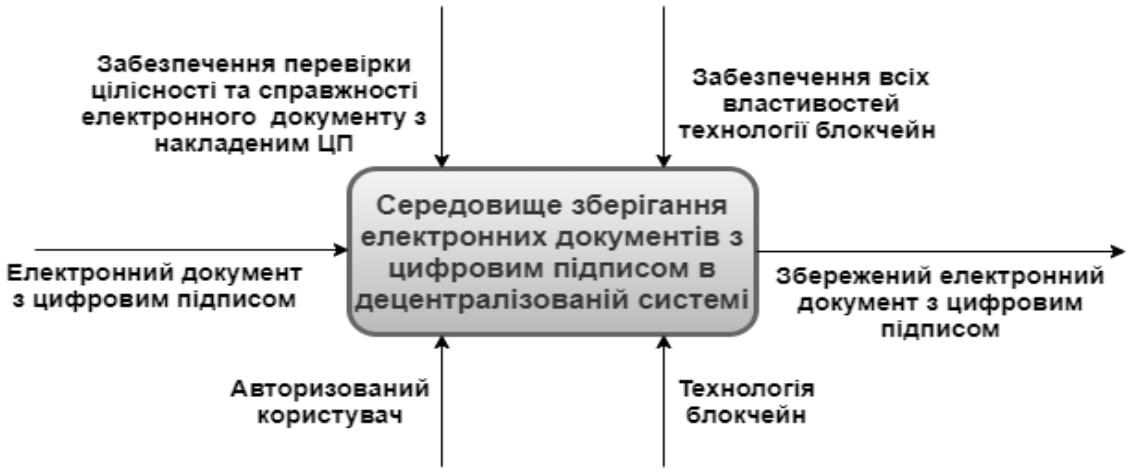


Рис. 4.3. Функціональна модель програмного забезпечення

Логічна структура програмного забезпечення (див. рис. 4.4):

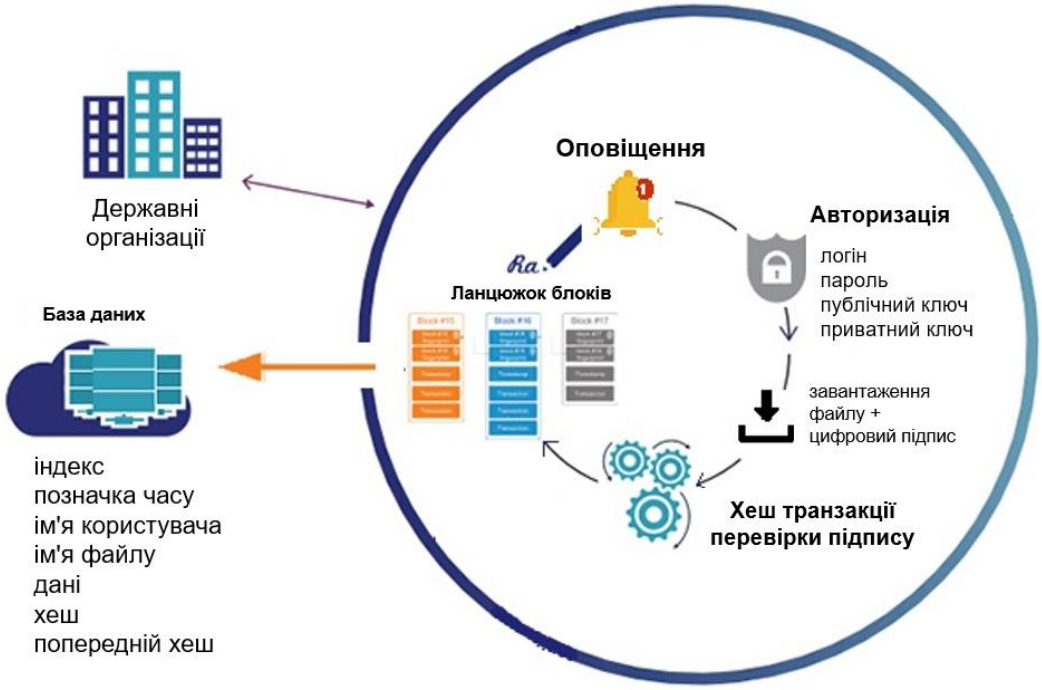


Рис. 4.4. Логічна структура програмного забезпечення

Крок 1: користувач входить до системи, використовуючи логін та пароль, як ідентифікатор;

Крок 2: генерування пари “публічний-приватний ключ” для можливості накладання ЦП;

Крок 3: підписання ЕД за допомогою сервісу накладання ЦП;

Крок 4: завантаження ЕД за допомогою наданого інтерфейсом інструменту;

Крок 5: надсилання підписаного ЕД для збереження на сервісі;

Крок 6: перевірка справжності підпису та побудова ланцюжка блоків;

Крок 7: оповіщення користувачів про здійснення нової транзакції.

4.2 Розробка програмного забезпечення

Для розробки програмного забезпечення для збереження електронних документів з цифровим підписом було обрано наступні технології: Docker, MySQL, JavaScript, Node.js, PHP, jQuery.

Docker – програмна платформа для швидкої розробки, тестування і розгортання додатків у контейнерах. Docker упаковує програмне забезпечення в стандартизовані блоки, які називаються контейнерами. Контейнери є методом віртуалізації на рівні операційної системи, відокремлюють додаток від ОС. Кожен контейнер включає все необхідне для роботи програми: бібліотеки, системні інструменти, код і середовище виконання. На відміну від звичайної віртуальної машини, контейнер не встановлює всередині себе операційну систему і не має власних дисків.

Компоненти Docker:

– Docker file – набір інструкцій, які повідомляють програмі про те, як створити образ. Операційна система визначає мови, змінні середовища, розташування файлів, мережеві порти та інші необхідні компоненти, дії контейнера після його випуску, а також систему, на якій буде базуватися контейнер;

- Docker image – файл, який містить специфікації програмних компонентів, які виконуються в контейнері;
- Docker run – команда, яка шукає образ і запускає контейнер на його основі. Кожен контейнер заснований на образі. Контейнери призначені для тимчасового використання, але їх можна зупинити і перезапустити, що призведе контейнер в той же стан, в якому його зупинили;
- Docker Hub – сховище для спільного використання і управління контейнерами, де можна знайти офіційні Docker images open-source проектів і вендорів, а також неофіційні образи. Можна завантажувати образи контейнерів з корисним кодом або завантажити в Hub свої власні, зробивши їх публічними або приватними. Також можна створити локальний реєстр Docker;
- Docker Engine – ядром Docker, базової клієнт-серверної технологією, яка створює і запускає контейнери.

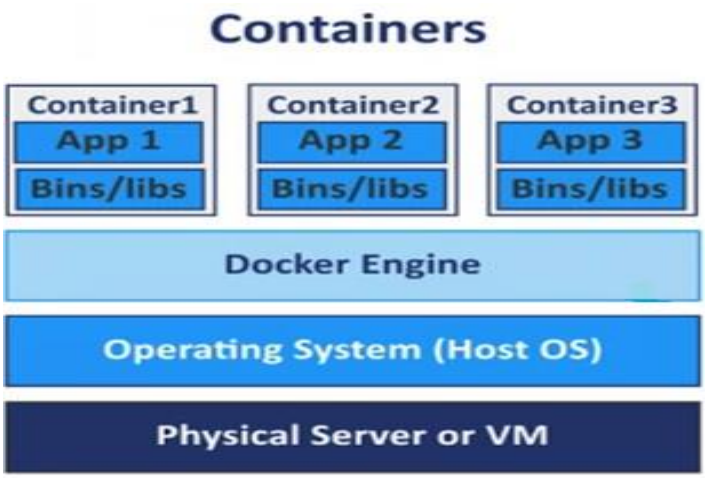


Рис. 4.5. Технологія контейнеризації Docker

Docker-compose.yml визначає правила створення та запуску багатоконтейнерних додатків Docker. В даному файлі описана структура середовища розробки та параметри, що необхідні для правильної роботи веб-додатків.

У розрізі blockchain, docker надає велику підтримку для швидкого запуску та запуску вузлів блокчейн без необхідності індивідуального налаштування кожної машини окремо. Також контейнеризація Docker надає додатковий захист та зменшує шанси втручання зловмисника в середовище.

Переваги:

- ізоляція. Контейнери мають власний виділений процесор, пам’ять та мережеві ресурси, тобто можна запускати кілька клієнтів блокчейн на одній машині. Кожен контейнер не може переглядати інші контейнери, поточні процеси та ресурси хоста, що зменшує ймовірність втручання зловмисників;
- економія ресурсів. Docker не є віртуальною машиною. Він взаємодіє напряму з ядром комп’ютера і при цьому ізолює програму на рівні процесу;
- висока швидкість розгортання;
- безпека та контроль. Пошкодження клієнта блокчейн, який працює всередині контейнера, обмежується лише цим контейнером та не впливає на інші контейнери. При видаленні docker-контейнеру неможливо втратити дані, так як docker виконує функцію сервісу та дані йому не належать;
- постійне розгортання. При оновленні версії клієнта або додаванні на машину нового клієнта блокчейн, створюється нове зображення контейнера, яке можна буде перенести у зовнішні сховища Docker;
- стандартизація середовищ. Є можливість відкату версії.

Docker створює велику кількість контейнерів, які необхідні для належної роботи програмного рішення системи електронного документообігу.

Головним аспектом безпеки є велика кількість нодів (вузлів), які є ідентичними. На нодах зберігаються дані та файли системи електронного документообігу. У випадку збереження даних лише на одному комп’ютері зловмиснику буде легко зламати систему, тому в системі передбачено

зберігання інформації на великій кількості нодів, що забезпечують децентралізоване зберігання даних в системі.

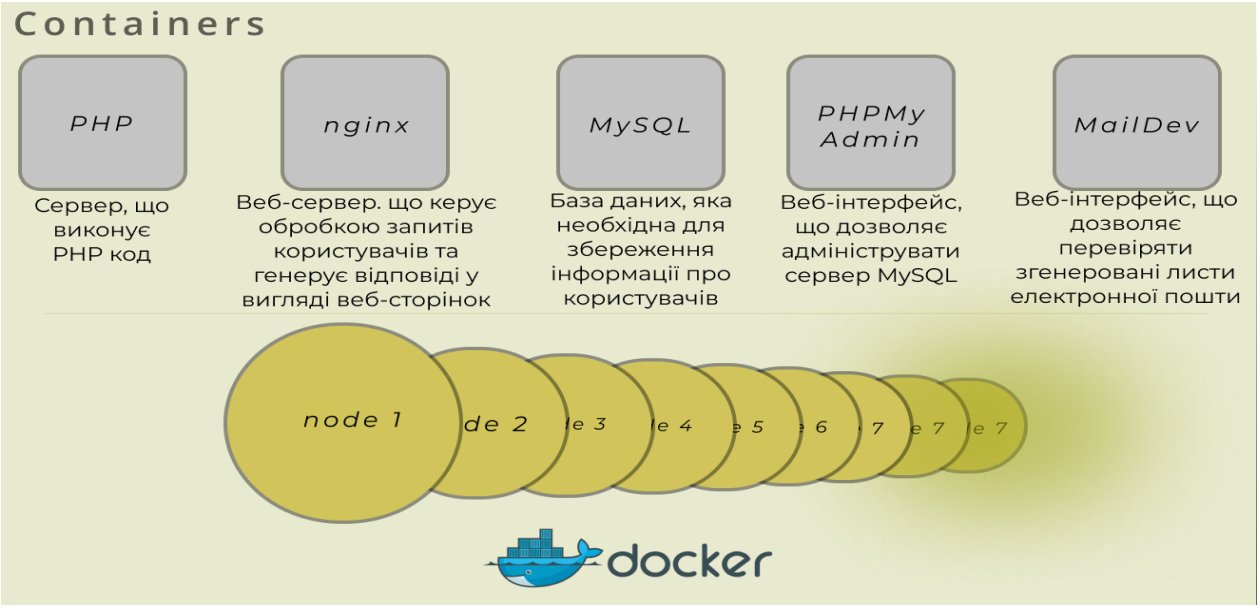


Рис. 4.6. Контейнери Docker

MySQL - це система керування базами даних, одна з найпопулярніших і найпоширеніших СУБД (система управління базами даних). MySQL відрізняться хорошою швидкістю роботи, надійністю, гнучкістю. Тобто це є спеціально створене сховище важливої інформації, невід'ємним атрибутом якого є зручний доступ до всіх даних, що зберігаються.

Переваги використання MySQL:

– простота в використанні;
– великий функціонал;
– безпека;
– масштабованість;
– швидкість.

JavaScript – об'єктно-орієнтована мова програмування, який додає інтерактивність веб-сайту. Зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. В JS скрипти запускаються в

браузері користувача, а не на сервері і зазвичай звертаються до бібліотек третьої сторони для забезпечення більш розширених функцій без потреби кожного разу писати цей код розробникам. Програми на мові JavaScript є автономними і вміщуються в документи, написані на мові HTML. Програма на мові JavaScript інтерпретується самим браузером при завантаженні документа, в який вміщений її код.

Node.js – кроссплатформенне середовище для виконання JavaScript. Це полегшене середовище, що використовується для розробки веб-додатків на стороні сервера. Використовують для створення будь-яких сервісів, де потрібен постійний обмін інформацією з користувачем. Node.js використовує керовану подіями неблокуючу модель введення-виведення, яка робить її придатною для додатків, що працюють з великими обсягами даних в реальному часі.

PHP – препроцесор гіпертексту (HTML), серверна мова програмування, скриптова, інтерпретована мова програмування. Тобто PHP є мовою програмування, яка створена для генерування HTML-сторінок на веб-сервері і роботою з базами даних. Дана мова спрямована на розробку веб-додатків та веб-сайтів. Дозволяє обробляти різні функції на стороні сервера, такі як збір даних форми, управління файлами на сервері, зміна баз даних та ін.

JQuery – швидка, невелика і багата можливостями JavaScript бібліотека. Вона дозволяє дуже просто робити такі речі як: обхід елементів або маніпуляція елементами HTML документа. JQuery допоможе спростити виконання складних завдань і прискорити розробку проекту.

Переваги JQuery:

- кросбраузерність;
- компактність коду;
- зручна робота з подіями і візуальними ефектами;
- велика кількість готових плагінів.

Враховуючи актуальність підібраних технологій дану систему можливо спрямувати в подальшому на вдосконалення.

4.3 Тестування системи електронного документообігу

При тестуванні програмного забезпечення було перевірено функціональність розроблювального програмного засобу відповідно до поставлених вимог СЕД. Для користувача була розроблена клієнтська частина, яка є веб-інтерфейсом. Задля зручної роботи достатньо встановленого веб-браузера.

За допомогою реалізації серверної частини таким чином, даний принцип дозволяє забезпечити належний рівень захищеності даних від змін. Маніпуляції з даними неможливо виконати без відома всієї системи.

Для початку роботи в системі електронного документообігу необхідно здійснити реєстрацію користувача. При реєстрації необхідно вказати логін, пароль, ім’я та електронну пошту. При реєстрації можливо згенерувати за допомогою реалізованого сервісу пару “публічний-приватний” ключ, що необхідно для підписання та перевірки транзакції. У випадку, якщо користувач має власні ключі, він може завантажити свій публічний ключ для підтвердження власника (див.рис.4.7).

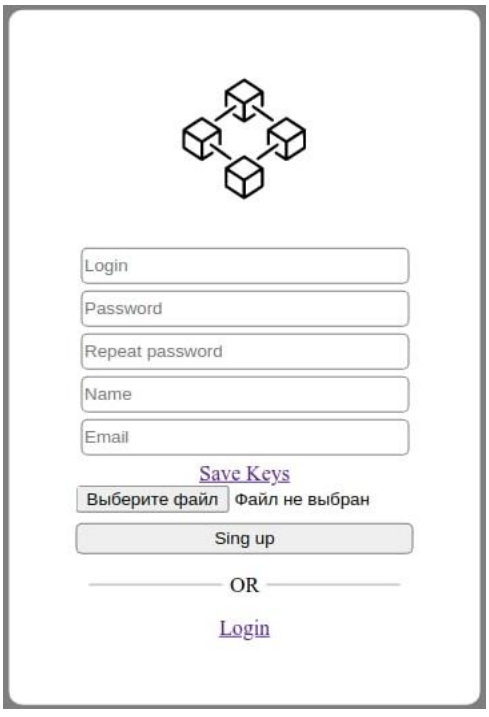


Рис. 4.7. Форма реєстрації

Дані користувача що здійснив реєстрацію зберігаються в базі даних, доступ до якої має лише адміністратор. Саме в базі даних адміністратор може надати або відхилити доступ до системи користувачу.

При спробі входу не підтвердженого користувача, він отримає наступне повідомлення (див.рис.4.8). Таким чином доступ до системи зможе отримати лише обмежене коло осіб.

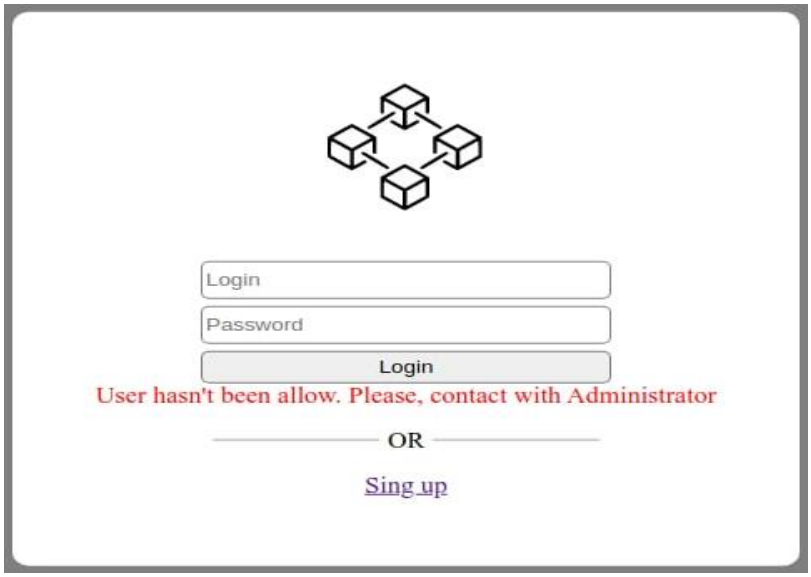


Рис. 4.8. Відмова доступу для користувача

При завантаженні файлу до системи користувач завантажує приватний ключ, який він зберігає в секреті для підписання (див.рис.4.9). Якщо користувач завантажує файл іншого формату підписання транзакції не відбудеться.

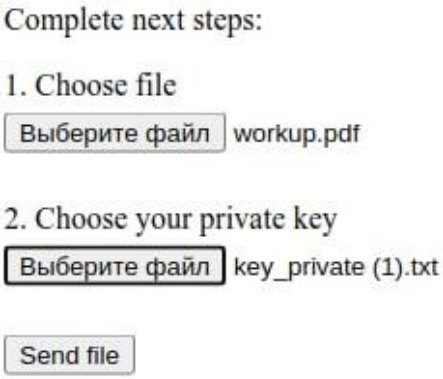


Рис. 4.9. Завантаження файлу

Цифровий підпис можна побачити натиснувши кнопку information (див.рис.4.10).

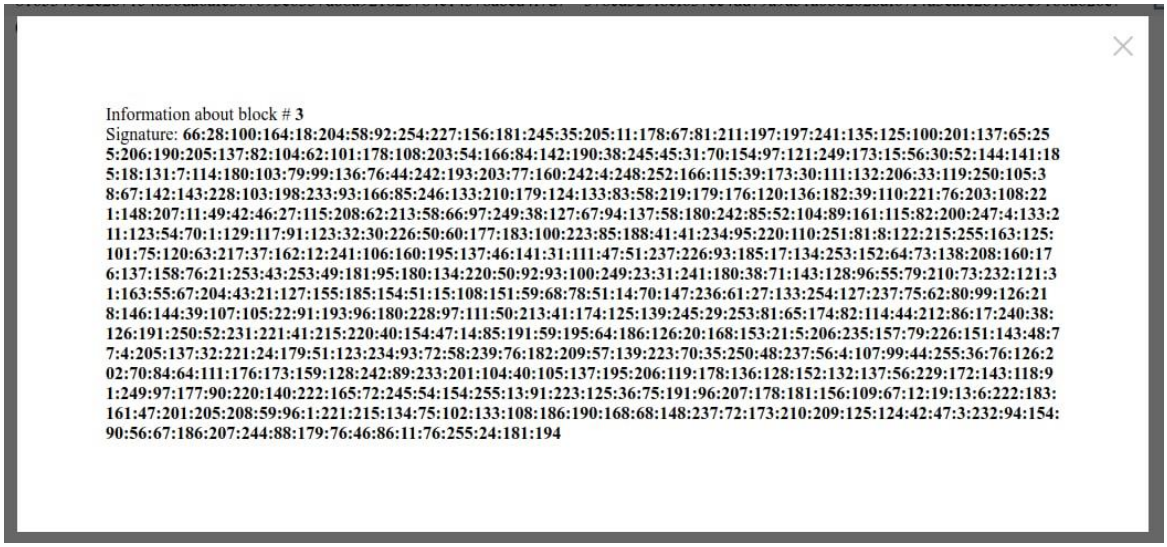


Рис. 4.10. Цифровий підпис

При правильній перевірці підпису індикатор має зелений колір, в іншому випадку – червоний. Для детального ознайомлення з документом користувач має змогу його завантажити. Ключовою особливістю системи є те, що будь-який користувач не може змінити стан транзакції, шляхом видалення чи редагування вже прийнятої транзакції. Простий графічний інтерфейс програми дозволяє простежити цілісність блоків та можливість верифікації підпису. Система є досить простою в використанні та може застосовуватись в державних структурах.

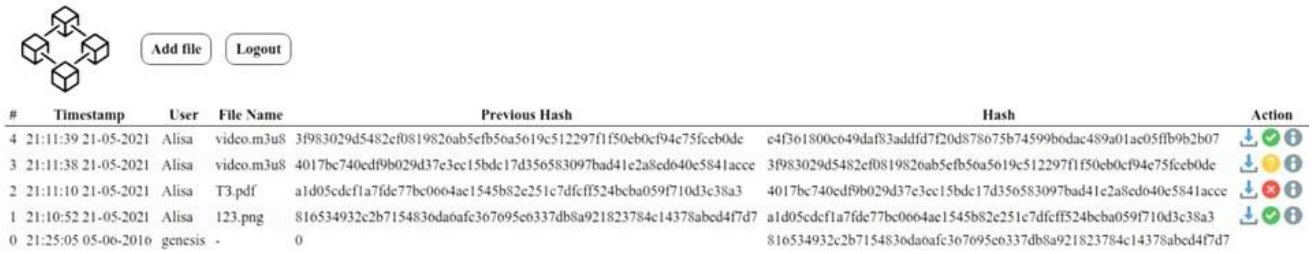


Рис. 4.11. Графічний інтерфейс СЕД

Висновки до Розділу 4

В даному розділі створено модель програмного забезпечення, описано технології та компоненти, які використовуються для реалізації системи, реалізовано систему електронного документообігу на основі технології blockchain. Також перевірено функціональність програмного засобу.

Система електронного документообігу цілком може застосовуватись в державних структурах, де критично необхідно надійне зберігання файлів.

ВИСНОВКИ

В даній роботі розглянуто основні поняття захисту інформації технології blockchain, звернено увагу на значимість ІБ, розглянуто особливості асиметричної криптографії, що є основою технології blockchain. Наведено особливості застосування цифрового підпису та хешування, яке забезпечує незмінність всього ланцюжка транзакцій blockchain.

Наведено визначення, класифікацію та особливості технології blockchain. Вказані основні структурні блоки та алгоритми досягнення консенсусу. З огляду на класифікацію, було обрано використання приватного типу блокчейн для застосування лише для обмеженого кола користувачів.

Також визначено вже існуючі системи зберігання електронних документів. Вказано властивості ефективної системи електронного документообігу, проаналізовано переваги та можливості технології blockchain для її реалізації.

В останньому розділі описано логічну структуру та основні компоненти, що необхідні для реалізації системи, застосовано середовище контейнеризації Docker для побудови системи електронного документообігу, наведено програмну реалізацію системи електронного документообігу з накладеним цифровим підписом на основі технології blockchain.

Мету роботи досягнуто, основні завдання виконано: досліджено принципи роботи технології blockchain; проаналізовано можливості технології blockchain для побудови системи електронного документообігу; розроблено програмний засіб системи електронного документообігу на основі технології blockchain. Матеріал даної роботи можна застосовувати в подальшому для розробки та впровадження системи електронного документообігу в межах організації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про інформацію: Закон України від 01.01.2017 № 48. База даних Законодавство України / ВР України. URL: https://zakon.rada.gov.ua/laws/show/2657-12 (дата звернення: 17.05.2021)
2. Богуш В. М., Кузин А. М. Інформаційна безпека від А до Я: 3000 термінів і понять: навчальний посібник. К : МОУ, 1999. 456 с
3. Белов Е.Б, Лось В.П, Мещеряков Р.В., Шелупанов А.А. Основы информационной безопасности: учебное пособие для вузов. М :Телеком, 2006. 544 с
4. Тарнавський Ю.А. Технології захисту інформації: навчальний посібник. К : КПІ ім. Ігоря Сікорського, 2018. 17 с
5. Положення про порядок здійснення криптографічного захисту інформації URL: https://zakon.rada.gov.ua/go/505/98 (дата звернення: 18.05.2021)
6. Криптографічні методи захисту інформації URL: https://tux.org.ua/kriptografichni-metodi-zahistu-informatsiyi (дата звернення: 19.05.2021)
7. Кукарін О.Б. Електронний документообіг та захист інформації : навчальний посібник. К:НАДУ, 2015. 18 с
8. Фороузан Б.А. Криптография и безопасность сетей: учебное пособие. М:БИНОМ, 2010. 396 с
9. Фороузан Б.А. Криптография и безопасность сетей: учебное пособие. М:БИНОМ, 2010. 393 с
10. K. Wüst and A. Gervais. Do you need a Blockchain. IACR Cryptol. ePrint Arch., 2017. 375 p
11. W. Meng et al.: When Intrusion Detection Meets Blockchain Technology: A Review. IEEE Access, 2018. 5 p

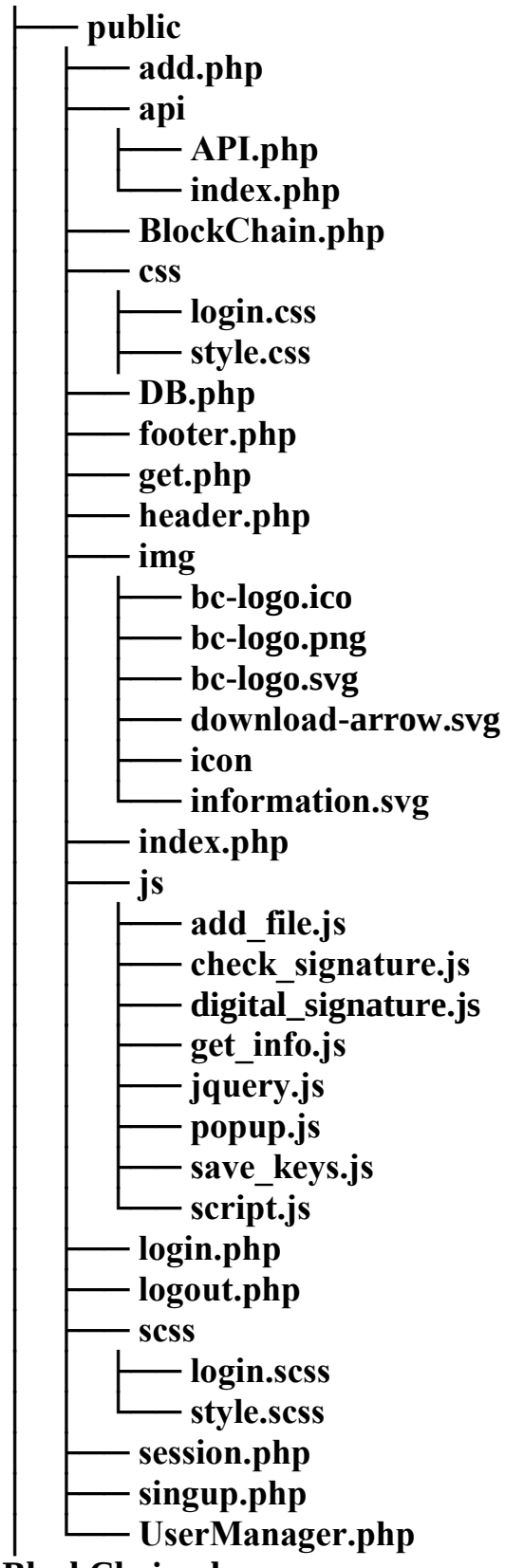
12. Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System URL: https://schbit.com/2018/03/18/bitcoin-robota-satoshi-nakamoto (дата звернення: 22.05.2021)
13. Класифікація блокчейн URL: https://polygant.net/ru/blog/vidy-blokchejna/ (дата звернення: 22.05.2021)
14. Dylan Yaga, Peter Mell, Nik Roby, Karen Scarfone. Blockchain Technology Overview. U.S. Department of Commerce, 2018. 15 p
15. Dylan Yaga, Peter Mell, Nik Roby, Karen Scarfone. Blockchain Technology Overview. U.S. Department of Commerce, 2018. 18 p
16. Cynthia Dwork, Moni Naor. Pricing via Processing or Combatting Junk Mail. Institute of Science, 1992. 3p
17. Rui Zhang, Rui Xue, and Ling Liu. Security and Privacy on Blockchain. ACM Comput. Surv, 2019. 19 p
18. Dylan Yaga, Peter Mell, Nik Roby, Karen Scarfone. Blockchain Technology Overview. U.S. Department of Commerce, 2018. 25 p
19. Про електронні документи та електронний документообіг: Закон України від 07.11.2018 № 36. База даних Законодавство України / ВР України. URL: https://zakon.rada.gov.ua/laws/show/851-15 (дата звернення: 25.05.2021).
20. Про електронні довірчі послуги: Закон України від 13.02.2020 № 440-IX База даних Законодавство України / ВР України. URL: https://zakon.rada.gov.ua/laws/show/2155-19 (дата звернення: 25.05.2021)
21. Чукут С.А., Буряченко К.О. Блокчейн чи система електронного документообігу: сучасні тенденції впровадження в органах виконавчої влади України: наукова стаття. К : КПІ ім. Ігоря Сікорського, 2018. 4 с
22. Фороузан Б.А. Криптография и безопасность сетей: учебное пособие. М:БИНОМ, 2010. 427 с

Додаток А

ВИХІДНИЙ КОД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ НА ОСНОВІ БЛОКЧЕЙН

Дерево проекту

```
app
    main.js
    node_modules
        body-parser
        crypto-js
        express
        ws
    package.json
    package-lock.json
docker
    db
        auto.cnf
        ca-key.pem
        ca.pem
        client-cert.pem
        client-key.pem
        dbname
        ib_buffer_pool
        ibdata1
        ib_logfile0
        ib_logfile1
        mysql
        performance_schema
        private_key.pem
        public_key.pem
        server-cert.pem
        server-key.pem
        sys
    mysql
        custom.cnf
    nginx
        nginx.conf
    php
        php.ini
docker-compose.yml
log
    access.log
    error.log
```




```
class BlockChain
{
   private $serverUrl;
    * BlockChain constructor.
   public function __construct()
```

```
    {
        $this->serverUrl = 'http://192.168.10.1';
    }
    public function addFile($file_name, $file_content, $signature, $user_id)
    {
        $this->addBlock($file_name, $file_content, $signature, $user_id);
    }
    /**
     * @param $data
     */
    public function addBlock($file_name, $data, $signature, $user_id)
    {
        $curl = curl_init();
        curl_setopt($curl, CURLOPT_URL, $this->serverUrl . '/mineBlock');
        curl_setopt($curl, CURLOPT_PORT, 3001);
        curl_setopt($curl, CURLOPT_POST, 1);
        curl_setopt($curl, CURLOPT_POSTFIELDS, '{"data" : "' . $data . '",
"fileName" : "' . $file_name . '", "signature" : "' . $signature . '", "user" : "' . $user_id .
'"}');
        $headers = array();
        $headers[] = 'Content-Type: application/json';
        curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
        curl_exec($curl);
        if (curl_errno($curl)) {
            echo 'Error:' . curl_error($curl);
        }
        curl_close($curl);
     */
    public function getFile($id)
    {
        $file_content = '';
        $file_name = '';
        $blocks = $this->getBlocks();
        if ($blocks) {
            $file_name = $blocks[$id]->fileName;
            $file_content = $blocks[$id]->data;
        }
        return ['fileName' => $file_name, 'fileData' => $file_content];
    }
    /**
     * @return mixed
     */
    public function getBlocks()
    {
```

```
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $this->serverUrl . '/blocks');
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_PORT, 3001);
    curl_setopt($curl, CURLOPT_POST, false);
    $result = curl_exec($curl);
    if (curl_errno($curl)) {
        echo 'Error:' . curl_error($curl);
    }
    curl_close($curl);
    return json_decode($result);
}
/**
 * Checking files with id
 * true - file created with current id
 * false - file didn't creat with current id
 * @param $id
 * @return bool
 */
public function checkID($id)
{
    foreach ($this->getBlocks() as $block) {
        $block_id = $this->getID($block->data);
        if ($id == $block_id) {
            return true;
        }
    }
    return false;
}
public function getHEX($valLength)
{
    $result = '';
    $moduleLength = 40;   // we use sha1, so module is 40 chars
    $steps = round(($valLength / $moduleLength) + 0.5);

    for ($i = 0; $i < $steps; $i++) {
        $result .= sha1(uniqid() . md5(rand() . uniqid()));
    }
    return substr($result, 0, $valLength);
}
/**
 * @return mixed
 */
public function getPeers()
```

```
    {
        $curl = curl_init();
        curl_setopt($curl, CURLOPT_URL, $this->serverUrl . '/peers');
        curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
        curl_setopt($curl, CURLOPT_PORT, 3001);
        curl_setopt($curl, CURLOPT_POST, false);
        $result = curl_exec($curl);
        if (curl_errno($curl)) {
            echo 'Error:' . curl_error($curl);
        }
        curl_close($curl);
        return json_decode($result);
    }
    public function __destruct()
    {
    }
}
```

Main.js

```
'use strict';
var CryptoJS = require("crypto-js");
var express = require("express");
var bodyParser = require('body-parser');
var WebSocket = require("ws");
var http_port = process.env.HTTP_PORT || 3001;
var p2p_port = process.env.P2P_PORT || 6001;
var initialPeers = process.env.PEERS ? process.env.PEERS.split(',') : [];
class Block {
    constructor(index, previousHash, timestamp, data, hash, fileName, user) {
        this.index = index;
        this.previousHash = previousHash.toString();
        this.timestamp = timestamp;
        this.data = data;
        this.hash = hash.toString();
        this.fileName = fileName;
        this.user = user;
    }
}
var sockets = [];
var MessageType = {
    QUERY_LATEST: 0,
    QUERY_ALL: 1,
    RESPONSE_BLOCKCHAIN: 2
};
var getGenesisBlock = () => {
```

```
        return new Block(0, "0", 1465154705, "my genesis block!!",
"816534932c2b7154836da6afc367695e6337db8a921823784c14378abed4f7d7","","g
enesis");
    };
    var blockchain = [getGenesisBlock()];
    var initHttpServer = () => {
        var app = express();
        app.use(bodyParser.json({limit: "4096mb", extended: true, parameterLimit:
4000000}));
        app.get('/blocks', (req, res) => res.send(JSON.stringify(blockchain)));
        app.post('/mineBlock', (req, res) => {
            var newBlock = generateNextBlock(req.body.data,
req.body.fileName,req.body.user);
            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));
    };
    var initP2PServer = () => {
        var server = new WebSocket.Server({port: p2p_port});
        server.on('connection', ws => initConnection(ws));
        console.log('listening websocket p2p port on: ' + p2p_port);
    };
    var initConnection = (ws) => {
        sockets.push(ws);
        initMessageHandler(ws);
        initErrorHandler(ws);
        write(ws, queryChainLengthMsg());
    };
    var initMessageHandler = (ws) => {
        ws.on('message', (data) => {
            var message = JSON.parse(data);
            console.log('Received message' + JSON.stringify(message));
            switch (message.type) {
```

```
                case MessageType.QUERY_LATEST:
                    write(ws, responseLatestMsg());
                    break;
                case MessageType.QUERY_ALL:
                    write(ws, responseChainMsg());
                    break;
                case MessageType.RESPONSE_BLOCKCHAIN:
                    handleBlockchainResponse(message);
                    break;
            }
        });
    };
    var initErrorHandler = (ws) => {
        var closeConnection = (ws) => {
            console.log('connection failed to peer: ' + ws.url);
            sockets.splice(sockets.indexOf(ws), 1);
        };
        ws.on('close', () => closeConnection(ws));
        ws.on('error', () => closeConnection(ws));
    };
    var generateNextBlock = (blockData,fileName,blockUser) => {
        var previousBlock = getLatestBlock();
        var nextIndex = previousBlock.index + 1;
        var nextTimestamp = new Date().getTime() / 1000;
        var nextHash = calculateHash(nextIndex, previousBlock.hash,
nextTimestamp, blockData, fileName, blockUser);
        return new Block(nextIndex, previousBlock.hash, nextTimestamp, blockData,
nextHash, fileName, blockUser);
    };
    var calculateHashForBlock = (block) => {
        return calculateHash(block.index, block.previousHash, block.timestamp,
block.data, block.fileName, block.user);
    };
    var calculateHash = (index, previousHash, timestamp, data, fileName, user) =>
{
        return CryptoJS.SHA256(index + previousHash + timestamp + data +
fileName + user).toString();
    };
    var addBlock = (newBlock) => {
        if (isValidNewBlock(newBlock, getLatestBlock())) {
            blockchain.push(newBlock);
        }
    };
    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(typeof (newBlock.hash) + ' ' + typeof
calculateHashForBlock(newBlock));
            console.log('invalid hash: ' + calculateHashForBlock(newBlock) + ' ' +
newBlock.hash);
            return false;
        }
        return true;
    };
    var connectToPeers = (newPeers) => {
        newPeers.forEach((peer) => {
            var ws = new WebSocket(peer);
            ws.on('open', () => initConnection(ws));
            ws.on('error', () => {
                console.log('connection failed')
            });
        });
    };
    var handleBlockchainResponse = (message) => {
        var receivedBlocks = JSON.parse(message.data).sort((b1, b2) => (b1.index -
b2.index));
        var latestBlockReceived = receivedBlocks[receivedBlocks.length - 1];
        var latestBlockHeld = getLatestBlock();
        if (latestBlockReceived.index > latestBlockHeld.index) {
            console.log('blockchain possibly behind. We got: ' + latestBlockHeld.index
+ ' Peer got: ' + latestBlockReceived.index);
            if (latestBlockHeld.hash === latestBlockReceived.previousHash) {
                console.log("We can append the received block to our chain");
                blockchain.push(latestBlockReceived);
                broadcast(responseLatestMsg());
            } else if (receivedBlocks.length === 1) {
                console.log("We have to query the chain from our peer");
                broadcast(queryAllMsg());
            } else {
                console.log("Received blockchain is longer than current blockchain");
                replaceChain(receivedBlocks);
            }
        } else {
```

```
            console.log('received blockchain is not longer than current blockchain. Do
nothing');
        }
    };
    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');
        }
    };
    var isValidChain = (blockchainToValidate) => {
        if (JSON.stringify(blockchainToValidate[0]) !==
JSON.stringify(getGenesisBlock())) {
            return false;
        }
        var tempBlocks = [blockchainToValidate[0]];
        for (var i = 1; i < blockchainToValidate.length; i++) {
            if (isValidNewBlock(blockchainToValidate[i], tempBlocks[i - 1])) {
                tempBlocks.push(blockchainToValidate[i]);
            } else {
                return false;
            }
        }
        return true;
    };
    var getLatestBlock = () => blockchain[blockchain.length - 1];
    var queryChainLengthMsg = () => ({'type': MessageType.QUERY_LATEST});
    var queryAllMsg = () => ({'type': MessageType.QUERY_ALL});
    var responseChainMsg = () =>({
        'type': MessageType.RESPONSE_BLOCKCHAIN, 'data':
JSON.stringify(blockchain)
    });
    var responseLatestMsg = () => ({
        'type': MessageType.RESPONSE_BLOCKCHAIN,
        'data': JSON.stringify([getLatestBlock()])
    });
    var write = (ws, message) => ws.send(JSON.stringify(message));
    var broadcast = (message) => sockets.forEach(socket => write(socket,
message));
    connectToPeers(initialPeers);
```

```
initHttpServer();
initP2PServer();
```

Digital_signature.js

```
async function generateKey() {
    const key = await window.crypto.subtle.generateKey({
            name: "RSASSA-PKCS1-v1_5",
            modulusLength: 4096,
            publicExponent: new Uint8Array([0x01, 0x00, 0x01]),
            hash: {
                name: "SHA-512"
            },
        },
        true,
        ["sign", "verify"]
    );
    return {
        privateKey: await window.crypto.subtle.exportKey(
            "jwk",
            key.privateKey,
        ),
        publicKey: await window.crypto.subtle.exportKey(
            "jwk",
            key.publicKey,
        ),
    };
}
async function sign(privateKeyJwk, message) {
    const privateKey = await window.crypto.subtle.importKey("jwk",
privateKeyJwk, {
        name: "RSASSA-PKCS1-v1_5",
        hash: {name: "SHA-512"},
    }, false, ['sign']);
    const data = new TextEncoder().encode(message);

    const signature = await window.crypto.subtle.sign({
            name: "RSASSA-PKCS1-v1_5",
        },
        privateKey,
        data,
    );
    // converts the signature to a colon seperated string
    return new Uint8Array(signature).join(':');
```

```
}
async function verify(publicKeyJwk, signatureStr, message) {
    const signatureArr = signatureStr.split(':').map(x => +x);
    const signature = new Uint8Array(signatureArr).buffer
    const publicKey = await window.crypto.subtle.importKey("jwk",
publicKeyJwk, {
        name: "RSASSA-PKCS1-v1_5",
        hash: {name: "SHA-512"},
    }, false, ['verify']);
    const data = new TextEncoder().encode(message);
    const ok = await window.crypto.subtle.verify({
            name: "RSASSA-PKCS1-v1_5",
        },
        publicKey,
        signature,
        data
    );
    return ok;
}
```

Check_signature.js

```
$(window).on('load', function() {
    $('.j-check-signature').click(function () {
        var button = $(this);
        var index = button.attr('data-index');
        var user_id = button.attr('data-userID');
        var signature = button.attr('data-signature');
        var public_key = '';
        var user_file = '';
        $.ajax({
            async: false,
            url: '../api/index.php',
            data: {
                get: 'publicKey',
                id: user_id,
            },
            type: 'GET',
            success: function (dataJSON) {
                data = JSON.parse(dataJSON);
                public_key = data.publicKey;
            }
        });
        $.ajax({
```

```
                async: false,
                url: '../api/index.php',
                data: {
                    get: 'fileData',
                    id: index,
                },
                type: 'GET',
                success: function (dataJSON) {
                    data = JSON.parse(dataJSON);
                    user_file = data.fileData;
                }
            });
            verify(public_key, signature, user_file).then(function (value) {
                console.log("Result: " + value);
                if(value === true) {
                    button.html('<svg version="1.1" width="20px" height="20px"
id="Capa_1" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0
367.805 367.805" style="enable-background:new 0 0 367.805 367.805;"
xml:space="preserve"><g><path style="fill:#3BB54A;"
d="M183.903,0.001c101.566,0,183.902,82.336,183.902,183.902s-82.336,183.902-
183.902,183.902 S0.001,285.469,0.001,183.903l0,0C-
0.288,82.625,81.579,0.29,182.856,0.001C183.205,0,183.554,0,183.903,0.001z"/><po
lygon style="fill:#fafafa;" points="285.78,133.225 155.168,263.837 82.025,191.217
111.805,161.96 155.168,204.801 256.001,103.968"/></g></svg>');
                }
                else {
                    button.html('<svg width="20px" height="20px" viewBox="0 0 512
512" xmlns="http://www.w3.org/2000/svg"><path d="m256 0c-141.164062 0-256
114.835938-256 256s114.835938 256 256 256 256-114.835938 256-256-
114.835938-256-256-256zm0 0" fill="#f44336"/><path d="m350.273438
320.105469c8.339843 8.34375 8.339843 21.824219 0 30.167969-4.160157
4.160156-9.621094 6.25-15.085938 6.25-5.460938 0-10.921875-2.089844-
15.082031-6.25l-64.105469-64.109376-64.105469 64.109376c-4.160156 4.160156-
9.621093 6.25-15.082031 6.25-5.464844 0-10.925781-2.089844-15.085938-6.25-
8.339843-8.34375-8.339843-21.824219 0-30.167969l64.109376-64.105469-
64.109376-64.105469c-8.339843-8.34375-8.339843-21.824219 0-30.167969
8.34375-8.339843 21.824219-8.339843 30.167969 0l64.105469 64.109376
64.105469-64.109376c8.34375-8.339843 21.824219-8.339843 30.167969 0
8.339843 8.34375 8.339843 21.824219 0 30.167969l-64.109376 64.105469zm0 0"
fill="#fafafa"/></svg>');
                }
            });
        });
```

});

Docker-compose.yml

```
version: '3'
services:
   php:
      image: atillay/lemp-php:7.3 # Also available : atillay/lemp-php:7.2
      env_file:
         - .env
      volumes:
         - ./docker/php/php.ini:/usr/local/etc/php/php.ini
         - .:/var/www:cached
      networks:
         - blockchain-network
      links:
         - node1:node1
   nginx:
      image: atillay/lemp-nginx
      networks:
         - blockchain-network
      ports:
         - ${SERVER_PORT}:80
      volumes:
         - ./docker/nginx/nginx.conf:/etc/nginx/nginx.conf
         - ./log:/var/log/nginx
         - ./public:/var/www/public
      links:
         - node1:node1
   mysql:
      image: mysql:5.7
      environment:
         - MYSQL_ROOT_PASSWORD=StrongPassword
         - MYSQL_DATABASE=${DB_NAME}
         - MYSQL_USER=${DB_USER}
         - MYSQL_PASSWORD=${DB_PASSWORD}
      volumes:
         - ./docker/mysql/custom.cnf:/etc/mysql/conf.d/custom.cnf
         - ./docker/db:/var/lib/mysql:cached
      networks:
         - blockchain-network
   phpmyadmin:
      image: phpmyadmin/phpmyadmin
```

```
    networks:
      - blockchain-network
    ports:
      - ${PMA_PORT}:80
    environment:
      - PMA_HOST=${DB_HOST}
maildev:
    image: djfarrelly/maildev
    networks:
      - blockchain-network
    ports:
      - ${MAILDEV_PORT}:80
node1:
    build:
      context: ./.docker/app/
      dockerfile: Dockerfile
    volumes:
      - ./app/:/naivechain
    networks:
      - blockchain-network
    ports:
      - "3001:3001"
node2:
    environment:
      - PEERS=ws://node1:6001
    build:
      context: ./.docker/app/
      dockerfile: Dockerfile
    volumes:
      - ./app/:/naivechain
    networks:
      - blockchain-network
    ports:
      - "3002:3001"
    links:
      - node1:node1
node3:
    environment:
      - PEERS=ws://node2:6001
    build:
      context: ./.docker/app/
      dockerfile: Dockerfile
    volumes:
      - ./app/:/naivechain
```

```
        networks:
            - blockchain-network
        ports:
            - "3003:3001"
        links:
            - node2:node2
networks:
    blockchain-network:
        driver: bridge
        ipam:
            driver: default
            config:
                - subnet: 192.168.10.0/24
```

Save_keys.js

```
$(window).on('load', function() {
    $('.j-save-keys').click(function (){
        saveKeys();
    });
    function saveKeys() {
        var keys = generateKey();
        keys.then(function (value) {
            var privateKey = value.privateKey;
            var publicKey = value.publicKey;
            download(JSON.stringify(privateKey), 'key_private', 'text/plain');
            download(JSON.stringify(publicKey), 'key_public', 'text/plain');
        });
    }
    function download(content, fileName, contentType) {
        var a = document.createElement("a");
        var file = new Blob([content], {type: contentType});
        a.href = URL.createObjectURL(file);
        a.download = fileName;
        a.click();
    }
});
```