

Основи криптовалют і блокчейн-технологій: від теорії до практики

Вітаємо вас на всеохоплюючому курсі з основ криптовалют та блокчейн-технологій! Ми розглянемо фундаментальні концепції криптопростору з особливим акцентом на Bitcoin – першій та найвпливовішій криптовалюті світу.

Наш курс допоможе вам зрозуміти ключові переваги та відмінні риси блокчейну порівняно з традиційними системами баз даних. Починаючи з криптовалют як перших застосувань блокчейну, ми поступово перейдемо до того, як ці технології можуть вирішувати нові проблеми в різних галузях.

 **by o d**

Структура курсу: шість модулів для повного розуміння



Огляд Bitcoin на високому рівні

Основні властивості та принципи роботи децентралізованої валюти



Історія блокчейну

Від руху Шифропанків до корпоративного використання



Технічний огляд механіки Bitcoin

Глибокий аналіз технологій і оптимізацій



Bitcoin у реальному житті

Гаманці, майнінг та практичне застосування




Теорія ігор і мережеві атаки

Уразливості та можливі загрози для мережі



Ethereum і смарт-контракти

Розширення можливостей децентралізованого майбутнього



Протокол і консенсус Bitcoin: загальний огляд

Ідентичність

Як створюються та ідентифікуються криптографічні адреси у мережі Bitcoin. Розглянемо принципи асиметричного шифрування та роль відкритих і закритих ключів.

Транзакції

Дослідження механізму проведення та верифікації транзакцій. Аналіз структури транзакцій та їх підписання за допомогою закритих ключів.

Ведення записів

Вивчення блокчейну як розподіленого реєстру даних. Особливості формування блоків та їх зв'язування через хеш-функції.

Консенсус

Алгоритми досягнення узгодженості в розподіленій мережі без центрального органу. Механізми Proof of Work та їх важливість для безпеки системи.

Фундаментальні концепції централізованих і децентралізованих валют

Централізовані валюти

Традиційні фіатні валюти, якими керують центральні банки та державні органи. Вони мають фізичну форму або електронні записи у централізованих базах даних.

- Контролюються урядом
- Підкріплені державною гарантією
- Вразливі до інфляції та політичного впливу
- Залежать від довіри до центрального органу

Децентралізовані валюти

Криптовалюти, що функціонують на основі блокчейн-технології без єдиного контролюючого органу. Вони існують як записи в розподіленому реєстрі, який підтримується мережею учасників.

- Не мають центрального органу контролю
- Забезпечені криптографією та математикою
- Прозорі та стійкі до цензури
- Базуються на довірі до протоколу та коду

Історія блокчейну: від руху Шифропанків до JP Morgan Chase





Революційне значення Bitcoin у порівнянні з попередниками

Вирішення проблеми "подвійних витрат"

Bitcoin вперше ефективно вирішив проблему подвійного витрачання (double-spending) без центрального посередника за допомогою консенсусного механізму Proof of Work та блокчейн-технології.

Повна децентралізація

На відміну від попередніх спроб, Bitcoin досяг реальної децентралізації – система не має єдиної точки відмови і успішно функціонує без центрального органу контролю.

Програмована обмеженість

Максимальна кількість Bitcoin обмежена 21 мільйоном монет, що робить його дефіцитним ресурсом, подібним до золота, але з перевагами цифрового активу.

Перша функціонуюча екосистема

Bitcoin створив першу діючу екосистему з реальними користувачами, що забезпечило мережевий ефект і стимулювало подальший розвиток всієї індустрії криптовалют.

Bitcoin механіка і оптимізації: технічний огляд

Мережа Bitcoin

P2P архітектура з повними та легкими вузлами для розподіленої верифікації

Схеми хеш-зобов'язань

Криптографічні механізми для забезпечення незмінності даних



Криптографія

Еліптичні криві та хеш-функції SHA-256 для безпеки транзакцій

Bitcoin Script

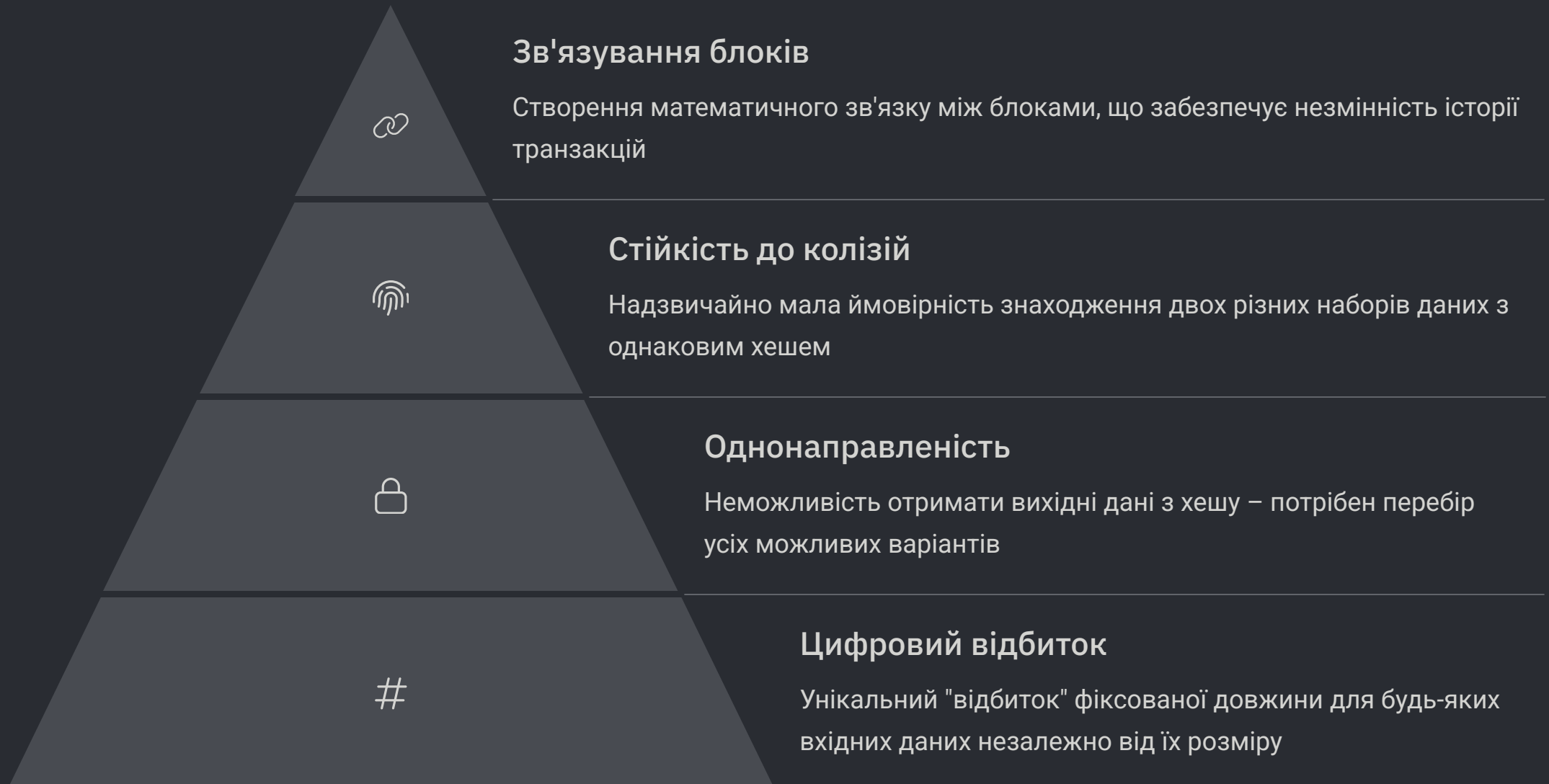
Проста мова програмування для визначення умов витрачання коштів

Конфіденційність

Псевдонімність та методи підвищення приватності транзакцій

Криптографічні хеш-функції: основа безпеки Bitcoin

Хеш-функції забезпечують критично важливі властивості, що роблять блокчейн Bitcoin надійним та безпечним.



Приклади хеш-функції. Як це працює

Хеш-функції забезпечують критично важливі властивості, що роблять блокчейн Bitcoin надійним та безпечним.

Приклад 1: Вхідні дані → Хеш

"Привіт" →

5dfac8d173362e5eef138dda6d1
530118b58ea42

Приклад 2: Лавинний ефект

"Привіт!" →

a4dc86f00831453635ef070f1f19
0749a05a254c

Зміна одного символу
призводить до повністю
іншого хешу

Приклад 3: Однаковий розмір

Для будь-якого обсягу вхідних даних хеш SHA-256 завжди має довжину 256 біт



**HASH
FUNCTION
VISUALIZATION**

SHA-256

Secure Hash Algorithm 256-bit (SHA-256): хеш-функція, що генерує вихід фіксованого розміру 256 біт (32 байти) незалежно від розміру вхідних даних. У Bitcoin використовується подвійний SHA-256 (хешування двічі) для посилення безпеки.

Характеристики SHA-256

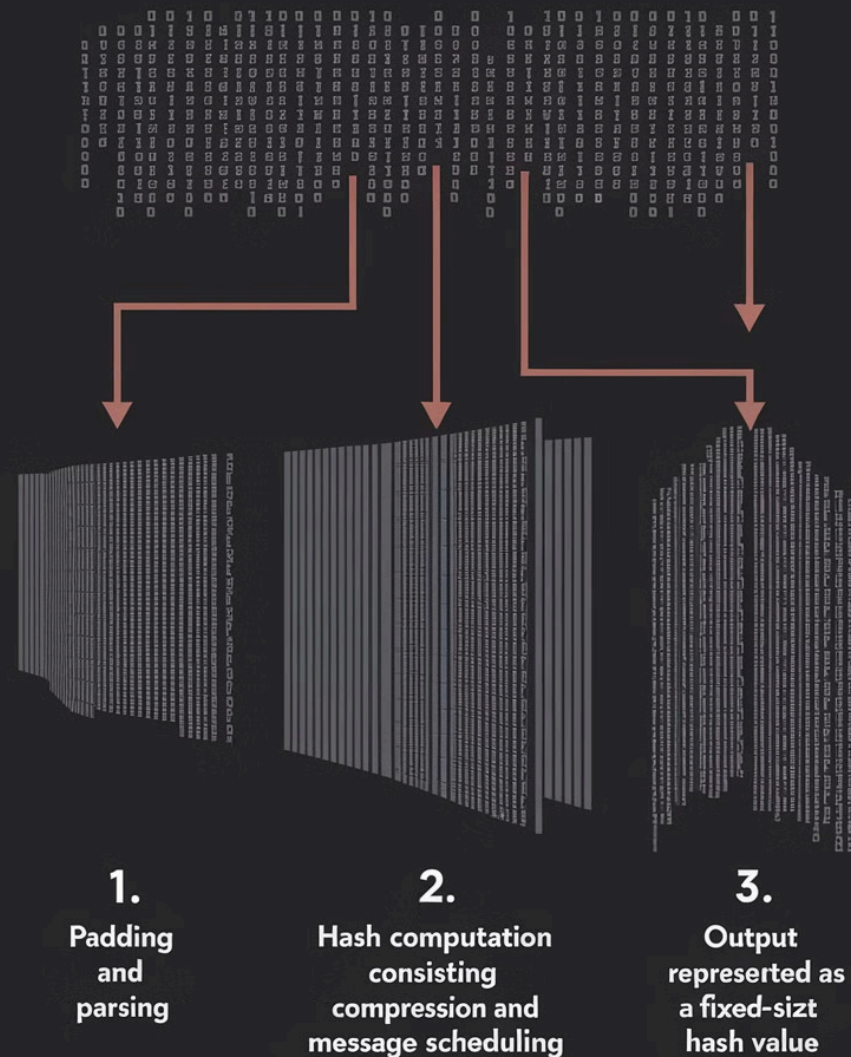
- Фіксований розмір виходу: 256 біт
- Подвійне хешування в Bitcoin
- Висока криптографічна стійкість

Застосування в Bitcoin

- Створення адрес гаманців з публічних ключів
- Обчислення хешу блоку для процесу майнінгу
- Забезпечення захисту Merkle Tree для транзакцій

Майнери мають знайти попсе, який при додаванні до заголовка блоку створює хеш, менший за цільове значення — основа механізму Proof-of-Work в Bitcoin.

SHA-256 Hash Function



Цифровий підпис. Пара ключів

Цифровий підпис у блокчейні забезпечує автентифікацію та цілісність транзакцій, гарантуючи, що тільки власник приватного ключа може розпоряджатися своїми активами.

Публічний ключ

Загальнодоступний ключ, який використовується для перевірки цифрового підпису та шифрування повідомлень.

- Поширюється вільно
- Математично пов'язаний з приватним ключем
- Неможливо вивести приватний ключ з публічного

Приватний ключ

Секретний ключ, що використовується для створення цифрового підпису та розшифрування повідомлень.

- Зберігається в таємниці
- Використовується для підписання транзакцій
- Втрата означає втрату доступу до коштів



Ключі. Математичний принцип 🧠 ✨

🔒 Що таке асиметрична криптографія?

🧩 Аналогія — замок і чарівний ключ

Уяви собі чарівний замок, який має:

- один ключ, який ЗАМИКАЄ замок — це **публічний ключ**
- і інший, **секретний ключ**, який ВІДМИКАЄ замок — це **приватний ключ**

📧 Ти можеш роздати всім замки (публічні ключі), щоб люди могли надсилати тобі "пошту", замикаючи її в коробку.

🔒 Але відкрити ці коробки можеш лише ти, бо в тебе є **приватний ключ**, який ні в кого більше немає.

🔄 А чому не можна просто "відгадати" приватний ключ?

Бо ці замки зроблені за **особливим математичним принципом**:

- ЛЕГКО зробити замок (з публічного ключа)
- але **майже неможливо** зробити ключ назад — це якби ти намагався розкласти кекс назад на яйце, муку і молоко 🍰🥚 — дуже складно.

🧠 Підсумок

🔒 **Асиметрична криптографія**
Це замок з одним ключем для замикання, і іншим — для відкривання

🌐 **Публічний ключ**
Відкритий, його можуть бачити всі

🌀 Еліптичні криві — магічне колесо



Тепер уяви:

- У нас є **магічне колесо з точками** (це еліптична крива)
- Ми крутимо його певну кількість разів (наприклад, 123456 разів) — і потрапляємо в точку на цьому колесі.

🔢 Це число (скільки разів крутимо) — **приватний ключ**

📍 Точка, куди прийшли — **публічний ключ**

🔒 Але! Якщо я бачу, куди ти прийшов (публічний ключ), я не знаю, **скільки разів** ти крутив колесо — надто багато варіантів!

🔑 **Приватний ключ**
Секретний, він зберігається в тебе

🕒 **Секрет неможливо вгадати**
Бо це як намагатись зібрати яйце з яєчні 🔍

Реалізація в коді

⚙ Як це виглядає на JavaScript

📦 1. Встановлення бібліотеки:

Для Node.js просто напишіть:

```
npm install elliptic
```

📄 Код: створення підпису та перевірка

```
const EC = require('elliptic').ec;
const sha256 = require('crypto-js/sha256');

// 1. Вибираємо тип кривої (таку ж використовує Bitcoin)
const ec = new EC('secp256k1');

// 2. Створюємо пару ключів
const key = ec.genKeyPair();

// 3. Наше повідомлення
const message = "Я перевів 1 BTC Олегу";

// 4. Робимо хеш повідомлення
const msgHash = sha256(message).toString();

// 5. Підписуємо хеш нашим приватним ключем
const signature = key.sign(msgHash);

// 6. Отримуємо публічний ключ для перевірки
const publicKey = key.getPublic('hex');

// 7. Перевіряємо чи підпис справжній
const isValid = ec.keyFromPublic(publicKey, 'hex').verify(msgHash, signature);

console.log("Підпис дійсний?", isValid); //
```

✅ true

🚫 Що буде, якщо хтось змінить повідомлення?

```
const fakeMessage = "Я перевів 100 BTC Олегу";
const fakeHash = sha256(fakeMessage).toString();

const isFakeValid = ec.keyFromPublic(publicKey, 'hex').verify(fakeHash, signature);
console.log("Підпис дійсний для зміненого повідомлення?", isFakeValid); //
```

❌ false

📌 Що робить кожна частина коду:

Функція	Призначення
ec.genKeyPair()	створює пару ключів: приватний і публічний
key.sign()	створює підпис для нашого повідомлення
key.getPublic()	показує публічний ключ іншим
verify()	перевіряє, чи підпис справжній

Асиметрична криптографія + цифровий підпис

🔑 Що таке асиметрична криптографія?

🔑 Аналогія: Чарівний штамп

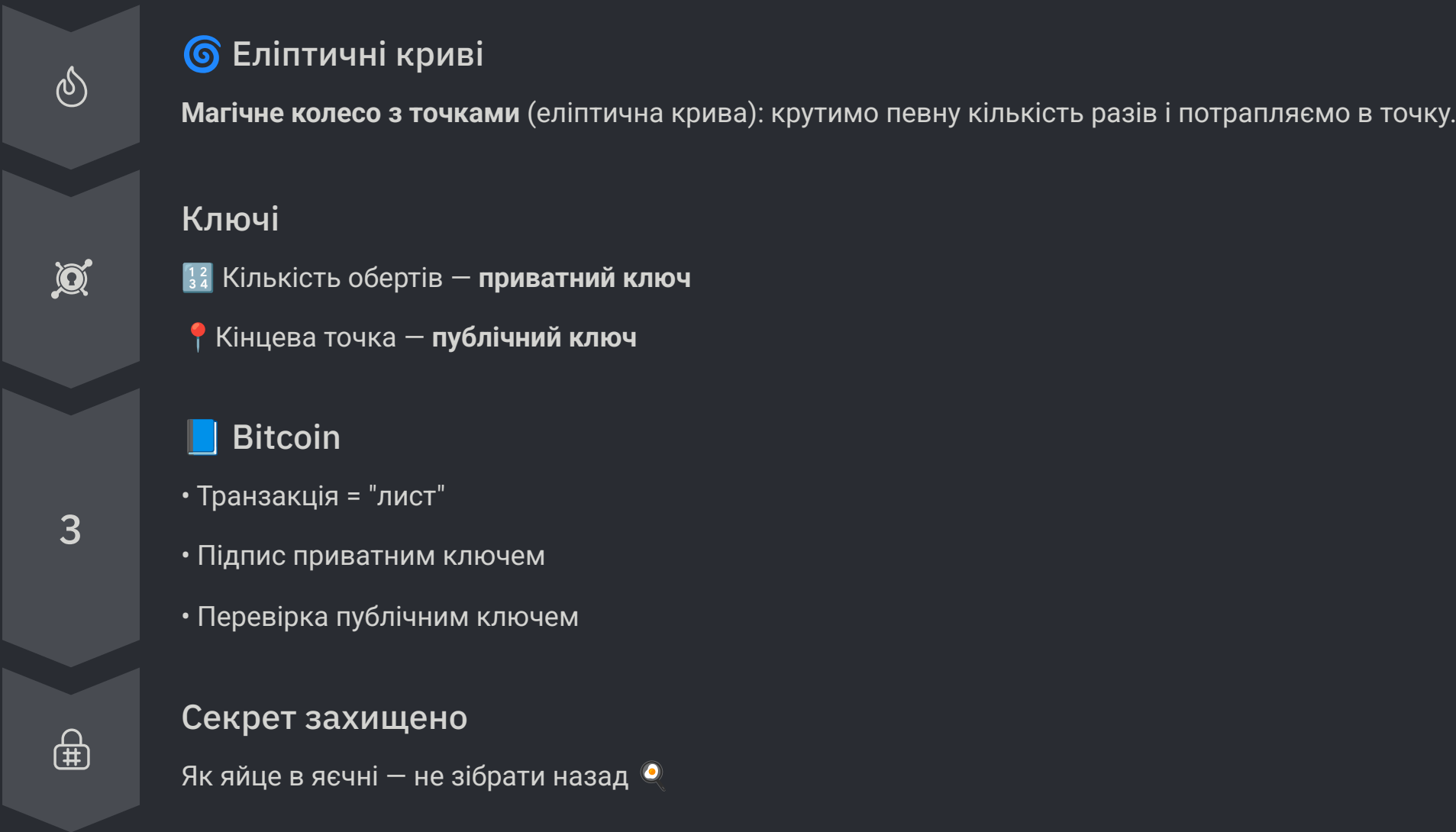
Уяви:

- Чарівний штамп (приватний ключ) для унікальних печаток.
- Печатку неможливо підробити.
- Спеціальна замкова щілина (публічний ключ) для перевірки:

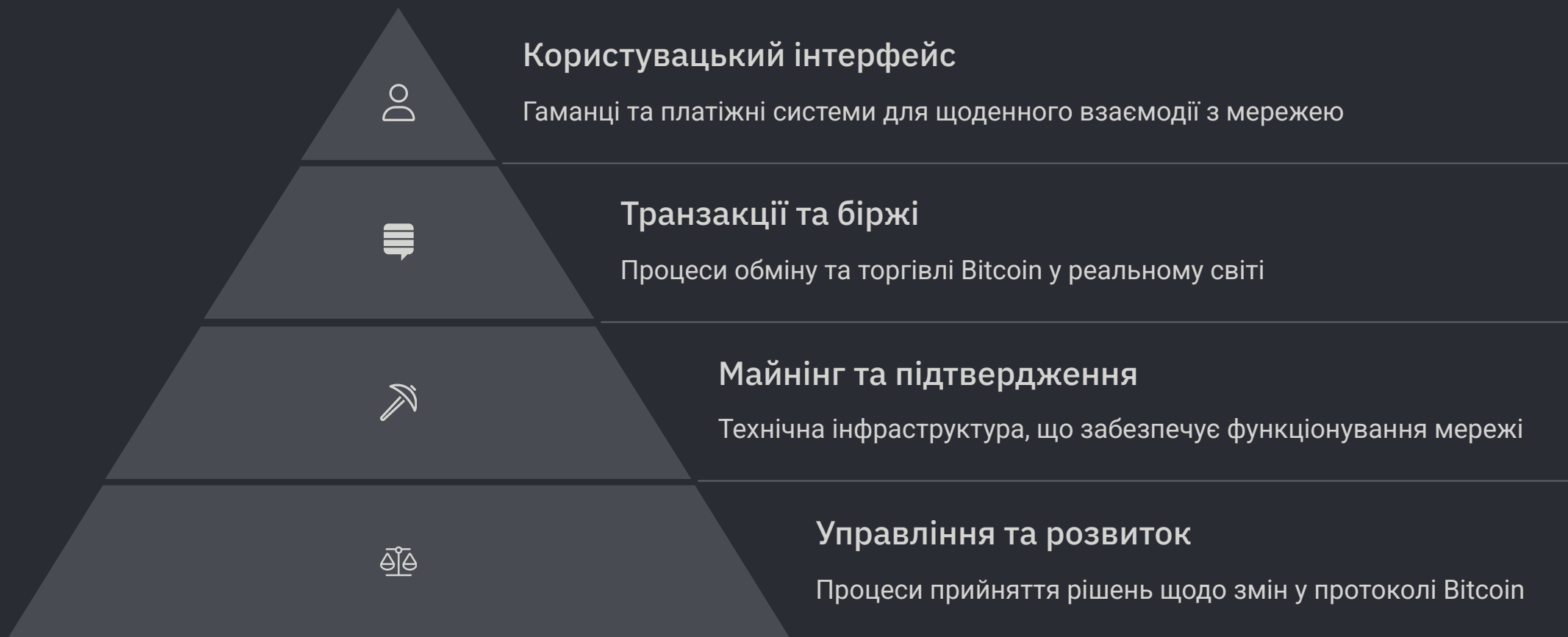
🟢 "Лист дійсно від Олександра"

🔴 "Підпис фальшивий"

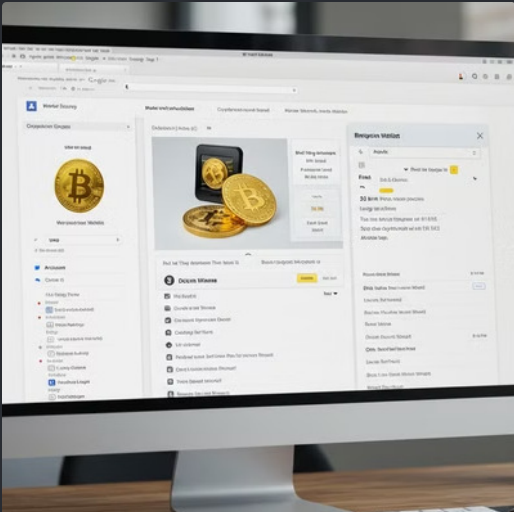
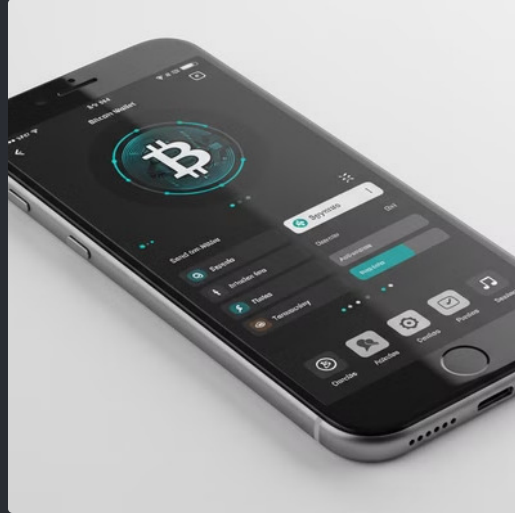
Як працює асиметрична криптографія в Bitcoin:



Bitcoin у реальному житті: гаманці, майнінг та транзакції



Типи Bitcoin-гаманців та їх механіка



Bitcoin-гаманці – це програмні або апаратні засоби, що дозволяють користувачам взаємодіяти з блокчейном Bitcoin. Залежно від потреб у безпеці та зручності, користувачі можуть обрати апаратні гаманці (найбезпечніші, але менш зручні), мобільні додатки (баланс безпеки та мобільності), настільні програми (для регулярного використання), паперові гаманці (для довгострокового зберігання) або веб-гаманці (найзручніші, але найменш безпечні).

Процес майнінгу Bitcoin: від транзакції до блоку



Створення транзакцій

Користувачі створюють транзакції з адресами відправника та отримувача, сумою та комісією. Кожна транзакція має цифровий підпис, який підтверджує право на використання монет.



Передача в мемпул

Підписані транзакції потрапляють у мережу Bitcoin. Кожен вузол перевіряє їх правильність та зберігає в своєму локальному сховищі (мемпулі) до моменту обробки.



Вибір транзакцій майнером

Майнери відбирають транзакції для нового блоку, віддаючи перевагу тим, що мають вищу комісію. Вони також додають спеціальну транзакцію з винагородою для себе.




Рішення криптографічної головоломки

Спеціальне обладнання перебирає мільйони варіантів, щоб знайти правильне число (хеш). Це схоже на лотерею, яка в середньому виграється раз на 10 хвилин.



Додавання блоку до ланцюга

Коли головоломка вирішена, новий блок відправляється всім учасникам мережі. Вони перевіряють його правильність і додають до свого блокчейну. Після цього майнери починають роботу над наступним блоком.



Теорія ігор і мережеві атаки: як зруйнувати Bitcoin



Атака "51%"

Якщо зломисник контролює більше 50% обчислювальної потужності мережі, він може змінювати історію транзакцій та блокувати нові транзакції. Ця атака стає все більш складною і дорогою з ростом мережі Bitcoin.



Атака подвійного витрачання

Зломисник намагається двічі витратити ті самі кошти, створюючи конфліктуючі транзакції та маніпулюючи підтвердженнями в мережі. Ризик зменшується з кожним додатковим підтвердженням блоку.



Мережеві атаки

DDoS-атаки на ключові вузли, атаки Sybil (створення багатьох фіктивних вузлів), атаки розщеплення мережі та атаки на систему DNS можуть тимчасово порушити роботу мережі Bitcoin.



Атака "Голдфінгер"

Атака з метою знищення або значного знецінення Bitcoin, мотивована зовнішніми економічними інтересами, наприклад, захистом традиційних фінансових систем від конкуренції.

Зловмисні стратегії майнінгу для отримання прибутку

Тип атаки	Механізм дії	Рівень загрози	Контрзаходи
Selfish Mining	Майнери приховують знайдені блоки та публікують їх стратегічно, щоб отримати перевагу	Середній	Модифікації протоколу вибору ланцюжка, алгоритми виявлення
Fee Sniping	Ігнорування поточного блоку на користь майбутніх блоків з вищими комісіями	Низький	Часові обмеження для блоків, резервування комісій
Block Withholding	Участь у пулі без публікації знайдених блоків, зменшення прибутку пулу	Високий для пулів	Розширені системи винагороди, моніторинг діяльності
Канібалізація пулів	Переключення між різними пулами для максимізації прибутку за рахунок інших	Середній	Удосконалені системи розподілу винагород, більша прозорість

Ці стратегії демонструють, як економічні стимули можуть впливати на безпеку Bitcoin. Багато з цих атак залишаються теоретичними, оскільки вартість їх реалізації часто перевищує потенційний прибуток, особливо з ростом мережі.



Ethereum і смарт-контракти: розширення можливостей блокчейну



Віртуальна машина Ethereum (EVM)

Децентралізований комп'ютер, що виконує смарт-контракти. EVM забезпечує Тюрінг-повноту, дозволяючи створювати програми будь-якої складності в екосистемі Ethereum, на відміну від обмеженого Bitcoin Script.



Смарт-контракти

Самовиконувані контракти з умовами угоди, записаними в коді. Вони автоматично виконуються при досягненні визначених умов без необхідності втручання третіх сторін, забезпечуючи прозорість та незмінність угоди.



Децентралізовані додатки (DApps)

Програми, що використовують смарт-контракти для бізнес-логіки, взаємодіючи з блокчейном. DApps охоплюють різноманітні галузі: від фінансів (DeFi) та ігор до управління ланцюгом поставок і цифрової ідентифікації.

Відмінності між Bitcoin та Ethereum: моделі UTXO vs. Акаунти

Дві найпопулярніші блокчейн-платформи використовують принципово різні архітектурні підходи:

Bitcoin (UTXO модель)

Архітектура: Використовує модель UTXO (Unspent Transaction Output), де "монети" існують як невитрачені результати попередніх транзакцій.

- **Безпека:** Висока (9/10) - пріоритетний напрямок системи
- **Масштабованість:** Середня (6/10) - поступово покращується
- **Програмованість:** Низька (2/10) - обмежена функціональність
- **Швидкість транзакцій:** Низька (3/10) - повільніша обробка
- **Гнучкість застосувань:** Обмежена (4/10) - фокус на валютній функції

Призначений переважно як цифрова валюта з акцентом на безпеку та стабільність.

Ethereum (Модель акаунтів)

Архітектура: Використовує модель акаунтів, де кожен користувач має обліковий запис із балансом та станом.

- **Безпека:** Висока (8/10) - надійна, але складніша система
- **Масштабованість:** Середня (5/10) - викликає певні труднощі
- **Програмованість:** Дуже висока (9/10) - основна перевага
- **Швидкість транзакцій:** Середня (7/10) - швидша за Bitcoin
- **Гнучкість застосувань:** Дуже висока (9/10) - різноманітні можливості

Орієнтований на створення децентралізованої обчислювальної платформи з підтримкою розширеного функціоналу смарт-контрактів.