# Біткоїн та криптовалютні технології
# Лекція 9: Масштабування Біткоїна 1/2

Юрій Жикін

21 квітня, 2025

# Пропускна здатність транзакцій

- Блок кожні 10 хвилин (600 секунд).
- Середній розмір кожного блока - 1.5 Мб.
- Середній розмір транзакції - 500 байтів.
- Пропускна здатність

$$T = \frac{1.5 * 1024 * 1024}{500 * 600} \approx 5 \text{ т/с}$$

- Пропускна здатність Visa - приблизно 1700 т/с.

# Пропозиції щодо масштабування пропускної здатності

- **Збільшення розміру блока** - збільшення централізації; поточна швидкість зростання розміру ланцюга - 80 Гб/рік.
- **Збільшення частоти створення блоків** - збільшення централізації, менша стабільність мережі.
- **Оптимізація структури транзакцій** - обмежені можливості.
- **Протоколи другого рівня** - єдиний практичний підхід?

# Оптимізація структури транзакцій

- **SegWit** (“відділений свідок”).
- **Підписи Шнорра**.
- **Меркелізовані абстрактні синтаксичні дерева (MAST)**.
- **Taproot (Schnorr + MAST)**.

# SegWit 1/4

- Оновлення протоколу, яке було активоване у 2017 році та вирішило наступні проблеми:
  - **проблема модифікації транзакцій** - протокол дозволяв внесення змін до транзакції так, що змінюється її ідентифікатор, при цьому підпис залишається дійсним;
  - **оптимізація використання місця у блоці** - усунення потреби зберігати великий скрипт відмикання в блоці - його перенесено в окрему структуру **Witness**;
  - **майбутні оновлення** - SegWit запровадив простий і чіткий механізм оновлення транзакційного протоколу через **м'які розгалуження** (softforks).

# SegWit 2/4

- SegWit продовжив ідею P2SH (BIP-0016) - додаткові правила перевірки скриптів активуються при розпізнаванні певного патерну в скрипті.
- Основні компоненти транзакції тепер - це **входи**, **виходи** та **свідки** (**witnesses**) (один **свідок** на кожен **вхід**).
- Для кожного входу транзакції, якщо **скрипт замикання**, який *виконується*, виглядає як патерн **програми-свідка**:
  - декодується відповідний **свідок**,
  - перевіряється, що **програма-відмикання** відповідає хешу **свідка**,
  - **свідок** інтерпретується як **скрипт відмикання**.

# SegWit 3/4

- **P2WPKH** - оплата за хеш публічного ключа свідчення (pay-to-witness-public-key-hash)

  Замикання: `0 <20-байтовий-хеш-ключа>;`

  Відмикання: `;`

  Свідок: `<підпис> <публічний ключ>`

- **P2WSH** - оплата за хеш скрипта свідчення (pay-to-witness-script-hash)

  Замикання: `0 <32-байтовий-хеш-ключа>;`

  Відмикання: `;`

  Свідок: `0 <підпис1> 1 <пубключ1> <пубключ2> 2 CHECKMULTISIG`

- **P2SH-P2WPKH** - P2WPKH вкладений у P2SH за BIP16

  Замикання: `HASH160 <20-байтовий-хеш-скрипта> EQUAL;`

  Відмикання: `<0 <20-байтовий-хеш-ключа>>;`

  Свідок: `<підпис> <публічний ключ>`

- **P2SH-P2WSH** - P2WSH вкладений у P2SH за BIP16

  Замикання: `HASH160 <20-байтовий-хеш> EQUAL;`

  Відмикання: `<0 <32-байтовий-хеш-ключа>>;`

  Свідок: `0 <підпис1> 1 <пубключ1> <пубключ2> 2 CHECKMULTISIG`

# SegWit 4/4

- Свідки включаються в ланцюг через **кореневий хеш** (**witness root hash**), який записується в **скрипті замикання** породжуючої транзакції.
- **Кореневий хеш** - це *корінь мерклевого дерева* побудованого з **wtxid**:

  `txid`: `[nVersion][txins][txouts][nLockTime]`

  `wtxid`: `[nVersion][marker][flag][txins][txouts][witness][nLockTime]`

- Обмеження на розмір блока $(1,000,000)$ змінено наступним чином:
  - *BlockWeight* визначається як $BaseSize * 3 + TotalSize$;
  - *BaseSize* - це розмір блока в байтах при оригінальній серіалізації транзакцій без будь-яких даних свідків;
  - *TotalSize* - це розмір блока в байтах з транзакціями, серіалізованими згідно з BIP144, включаючи як базові дані, так і дані свідків;
  - Нова умова: $BlockWeight \leq 4,000,000$.

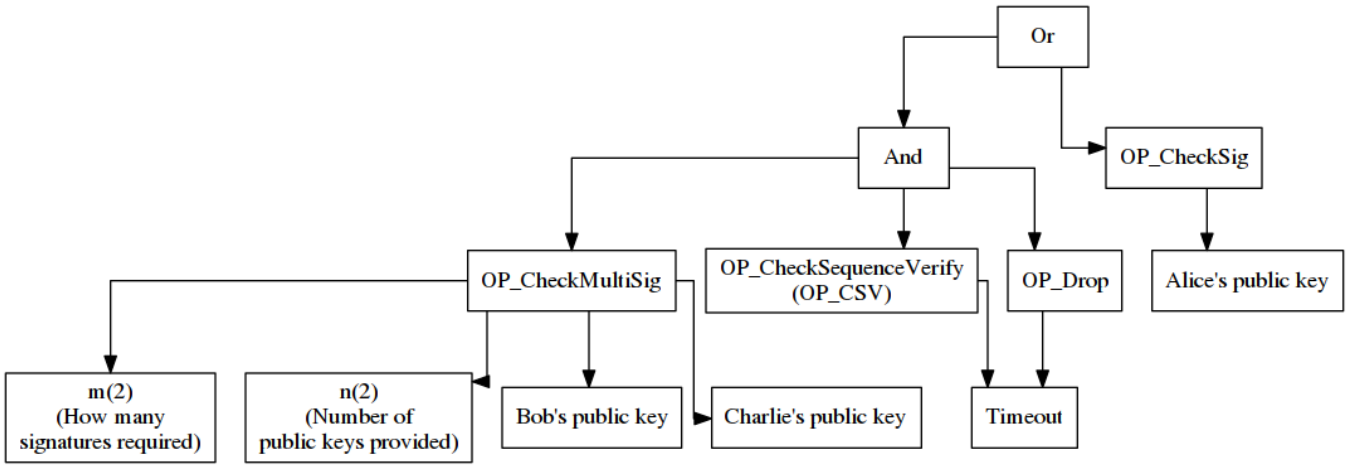
# Підписи Шнорра

- **Підпис Шнорра** - це альтернативна схема криптографічних підписів, яка забезпечує певні властивості, корисні для Біткоїн-системи, зокрема, **лінійність**, що дозволяє об'єднувати ключі та підписи:

$$key_x = key_1 + key_2 + \cdots + key_n$$
$$sig_x = sig_1 + sig_2 + \cdots + sig_n$$

# MAST і Taproot

- **Меркелізовані абстрактні синтаксичні дерева** (**MAST**) - це спосіб підтримки великих і складних скриптів шляхом побудови з них *мерклевого дерева* і відмикання за допомогою розкриття лише потрібного **шляху у дереві**, що підвищує конфіденційність.




- І **підписи Шнорра**, і варіант **MAST** є частиною м'якого розгалуження **Taproot**, який було активовано в основній мережі 14 листопада 2021 року.

# Рекомендовані ресурси

- **Bitcoin: A Work in Progress**
  - книга автора Сьорса Провоста (Sjors Provoost), одного з активних розробників Біткоїна
  - https://btcwip.com

# Кінець

Дякую за увагу!