Біткоїн і криптовалютні технології
Лекція 3: основи криптографії 2/2

Юрій Жикін

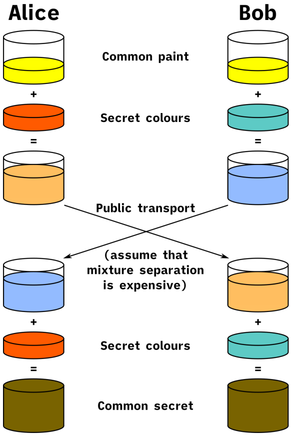
11 жовтня, 2022

Криптографія з публічним ключем: повторення

- **Криптографічна система з публічним ключем** або **асиметрична криптографічна система** - це криптографічна система, що використовує **пари** ключів:
  - **приватний ключ**, який має зберігатись в таємниці власником,
  - **публічний ключ**, який може бути публічно відомий всім.
- Ключова ідея криптографії з публічним ключем - це те, що будь-хто, хто знає публічний ключ, може "замкнути" якусь інформацію цим ключем таким чином, що лише власник приватного ключа може "відімкнути" її.
- Відкрито Ральфом Мерклом, Вітфілдом Діффі, Мартіном Гелманом та іншими у 1970-х.
- Чи не єдина причина, чому ми можемо робити хоч щось корисне в Інтернеті.

Ідея обміну ключами Діффі-Гелмана-Меркла

- Обмін ключами Діффі-Гелмана-Меркла інтуїтивно можна пояснити наступним прикладом:



Поля 1/2

- **Поле** - це множина елементів з визначеними операціями **додавання**, **віднімання**, **множення** та **ділення**, що задовільняє **аксіоми поля**:

  $\forall a, b, c : (a \star b) \star c = a \star (b \star c)$ - асоціативність для $+$ та $*$

  $\forall a, b : a \star b = b \star a$ - комутативність для $+$ та $*$

  $\exists e_+ = 0 : \forall a : e + a = 0 + a = a$ - адитивна одиниця

  $\exists e_* = 1 : \forall a : e * a = 1 * a = a$ - мультиплікативна одиниця

  $\forall a : \exists(-a) : a + (-a) = e_+ = 0$ - адитивна інверсія

  $\forall a \neq 0 : \exists(a^{-1}) : a * (a^{-1}) = e_* = 1$ - мультиплікативна інверсія

  $\forall a, b, c : a * (b + c) = (a * b) + (a * c)$ - дистибутивність $*$ над $+$

Поля 2/2

- Множина **раціональних чисел** $\mathbb{R}$ - це поле над звичайними операціями додавання та множення.
- В криптографії, ми зазвичай розглядаємо **скінченні поля Галуа простого порядку** над операціями модульної арифметики:

$$F_n = \mathbb{Z}/n\mathbb{Z} = 0, 1, ..., n-1$$

де n - це просте число; **така конструкція є полем тоді і тільки тоді, коли** n **- просте число**.

Групи 1/2

- **Група** - це множина елементів з визначеною бінарною операцією, яка комбінує два елементи у третій елемент групи таким чином, що виконуються три **аксіоми групи**

$$\forall a, b, c : (a \star b) \star c = a \star (b \star c) \text{ - асоціативність}$$
$$\exists e : \forall a : e \star a = a \text{ - існування одиниці}$$
$$\forall a : \exists b : a \star b = e \text{ - існування інверсії}$$

- **Породжуюча множина групи** - це підмножина елементів групи, така, що кожен елемент групи може бути представлений як комбінація скінченної кількості елементів цієї підмножини та їх обернених елементів.
- Група, що породжується одним елементом (який називається **генератором** і позначається літерою G), називається **циклічною**.

Групи 2/2

- Нехай $\mathbb{G}$ - це група. Нехай $a, b \in \mathbb{G}$. Позначимо операцію групи множенням, а її одиницю - 1. Нехай

$$b^k = a$$

- k, яке задовільняє попереднє рівняння, називається **дискретним логаритмом** a за основою b.
- Якщо ми позначимо операцію групи додаванням, а її одиницю - 0, позначення **дискретного логаритма** виглядатиме так

$$kb = a$$

- **Задача пошуку дискретного логаритма** або **DLOG-задача вважається дуже складною для деяких груп**.

Обмін ключами Діффі-Гелмана-Меркла 1/2

- Реалізації протоколу ДГМ базуються на наступному спостереженні, записаному адитивно

$$
\begin{aligned}
A &= aG(= G + G + \ldots + G) \\
B &= bG \\
bA &= b(aG) = (ba)G = (ab)G = a(bG) = aB
\end{aligned}
$$

або мультиплікативно

$$
\begin{aligned}
A &= G^a(= G * G * \ldots * G) \\
B &= G^b \\
A^b &= (G^a)^b = G^{(ab)} = G^{(ba)} = (G^b)^a = B^a
\end{aligned}
$$

Обмін ключами Діффі-Гелмана-Меркла 2/2

- Найпростіша реалізація протоколу ДГМ (як описано у статті) використовує **мультиплікативну групу цілих чисел по модулю** p, де p - **просте число**, а g - **примітивний корінь по модулю** p.
- Приклад ДГМ з малими числами:
  - Еліс та Боб домовились використовувати числа по модулю $p = 23$ та основу $G = 5$.
  - Еліс обирає таємне число $a = 4$ і надсилає Бобу

$$A = G^a \pmod{p} = 5^4 \pmod{23} = 4$$

  - Боб обирає таємне число $b = 3$ і надсилає Еліс

$$B = G^b \pmod{p} = 5^3 \pmod{23} = 10$$

  - Еліс обчислює

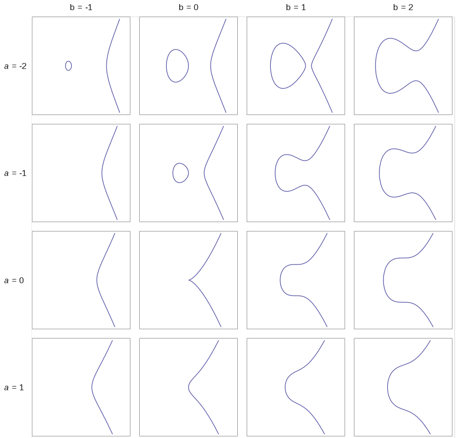
$$s = B^a \pmod{p} = 10^4 \pmod{23} = 18$$

  - Боб обчислює

$$s = A^b \pmod{p} = 4^3 \pmod{23} = 18$$

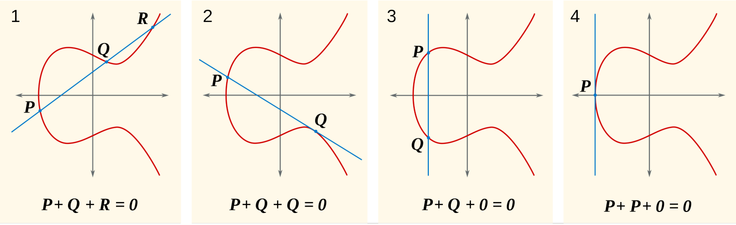
- **Еліптичні криві** - це алгебраїчні структури, що описуються рівняннями, які мають форму:

$$y^2 = x^3 + ax + b$$



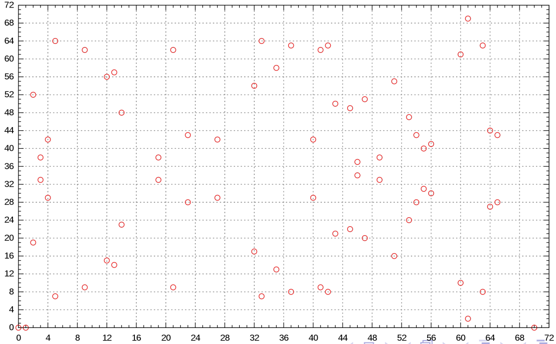
Еліптичні криві 2/3

- Еліптична крива, визначена над **полем** K, складається з точок в множині $K \times K$ і утворює **групу**.
- **Груповий закон**:
  - Якщо P та Q - дві точки кривої, тоді ми можемо однозначно описати третю точку кривої, $P + Q$, наступним чином. Спочатку, будуємо пряму, що перетинає криву у точках P та Q, яка в загальному випадку перетинає криву у ще одній точці, R. Тоді встановлюємо, що $P + Q$ - це точка $-R$, протилежна точці R відносно осі x.



Еліптичні криві 3/3

- **Еліптичні криві, визначені над скінченними полями простих порядків**, утворюють групи, що мають значно складнішу алгебраїчну структуру, і тому добре підходять для використання у криптографії, бо дозволяють використовувати значно **менші ключі**.
- Приклад еліптичної кривої, визначеної над скінченним полем ($y^2 = x^3 - x$ над F_{71}):



Криптографічні підписи

- **Схема криптографічних підписів** - це система для перевірки автентичності повідомлень.
- Схема криптографічних підписів складається з наступних трьох алгоритмів:
  - алгоритм генерації ключів *Gen*, який обирає приватний ключ випадковим чином з множини всіх можливих приватних ключів,
  - алгоритм підписування *Sign*, який, отримавши повідомлення та приватний ключ, створює підпис,
  - алгоритм перевірки підпису *Verify*, який, отримавши повідомлення, публічний ключ та сам підпис, або приймає, або відкидає підпис.
- Успішна перевірка підпису надає **дуже вагому** причину вірити, що дане повідомлення було автентифіковано власником відповідного приватного ключа.

DSA 1/2

- Еліс та Боб домовляються про параметри групи $(GROUP, G, n)$, де $GROUP$ - це група простого порядку n з генератором G.
- Еліс створює пару ключів, що складається з приватного цілого числа a, випадково вибраного на проміжку $[1, n-1]$ та публічного елемента групи $A = aG$.
- Для того, щоб підписати повідомлення m, Еліс
  - обчислює $e = HASH(m)$,
  - обирає **криптографічно безпечне випадкове ціле число** $k \in [1, n-1]$,
  - обчислює елемент групи $x = kG$,
  - обчислює $r = x \pmod{n}$,
  - обчислює $s = k^{-1}(e + ra) \pmod{n}$.
- Підпис - це пара значень (r, s); $(r, -s \pmod{n})$ - **також правильний підпис**.

DSA 2/2

- Для перевірки підпису (r, s) Боб
  - перевіряє, що r та s є цілими числами в проміжку $[1, n-1]$, інакше підпис недійсний,
  - обчислює $e = HASH(m)$,
  - обчислює $u_1 = es^{-1} \pmod{n}$ та $u_2 = rs^{-1} \pmod{n}$,
  - обчислює елемент групи $x = u_1 G + u_2 A$.
- Підпис дійсний, якщо $r \equiv x_1 \pmod{n}$ та недійсний у протилежному випадку:

$$
\begin{aligned}
u_1 G + u_2 A &= u_1 G + u_2 a G = (u_1 + u_2 a) G \\
&= (es^{-1} + rs^{-1} a) G = (e + ra) s^{-1} G \\
&= (e + ra)(a + ra)^{-1} k G = kG \\
&= r
\end{aligned}
$$

ECDSA

- **Алгоритм цифрових підписів на еліптичній кривій (Elliptic Curve Digital Signature Algorithm, ECDSA)** - це варіант алгоритму **DSA**, який використовує групу еліптичної кривої замість мультиплікативної групи.
- З **еліптичними кривими**, бітовий розмір приватного ключа, який вважається достатнім для **ECDSA**, приблизно **вдвічі більший** за бажаний рівень безпеки, тобто для отримання 128 бітів безпеки достатньо ключа розміром 256 бітів.
- З DSA, бітовий розмір ключа, що дає схожий рівень безпеки, становить 2048 або навіть 3072 бітів.

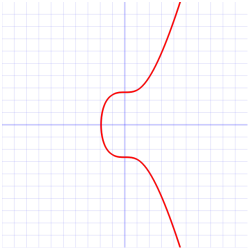
Еліптична крива SECP256K1 1/2

- Еліптична крива, що використовується у Біткоїні для підписування транзакцій, називається **secp256k1**.
- Ця еліптична крива визначена над скінченним полем F_p, де

$$p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$$

і описується рівнянням

$$y^2 = x^3 + 7$$



Еліптична крива SECP256K1 2/2

- Крива **secp256k1** побудована спеціальним невипадковим чином, що дозволяє використовувати високошвидкісну реалізацію операцій.
- На відміну від криптографічних кривих, що рекомендуються NIST, за винятком кривої **curve25519**), константи кривої **secp256k1** обрані передбачуваним чином, що значно зменшує ймовірність того, що в ній є "задні двері".
- **libsecp256k1** - це високооптимізована реалізація кривої **secp256k1**, що була виділена з бази коду проєкту Bitcoin Core в окремий проєкт:

  https://github.com/bitcoin-core/secp256k1
- Біткоїн використовує **secp256k1** як для традиційних підписів ECDSA, так і для підписів Шнора, які були впроваджені в рамках зміни проколу під назвою **Taproot** (англ. "стрижневий корінь"), яка була активована 14 листопада 2021 року.

Корисні ресурси

- **A Computational Introduction to Number Theory and Algebra** by Victor Shoup
  - https://shoup.net/ntb/

Кінець

Дякую за увагу!