Міністерство освіти і науки України
Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики
Кафедра прикладної математики

Бакалаврська робота

на тему:

Ефективність класичних алгоритмів криптографії на основі еліптичних кривих

Виконав студент групи ПМП-43:
Квас Кирил Олегович
спеціальність 113 — Прикладна математика

Науковий керівник:
доцент, кандидат фізико-математичних наук
Стягар Андрій Орестович

Рецензент:

_____

Львів — 2025

Зміст

Вступ **3**

1 Постановка задачі **4**
1.1 Описова постановка 4
1.2 Математична модель 4

2 Проблема дискретного логарифму **6**
2.1 Огляд головних криптографічних схем 6
2.1.1 Схеми на основі факторизації цілих чисел 6
2.1.2 Схеми на основі дискретного логарифму 6
2.1.3 Схеми на основі еліптичних кривих 6
2.1.4 Інші схеми відкритого ключа 7
2.2 Алгебраїчні структури в контексті дискретного логарифму 7
2.2.1 Групи 7
2.2.2 Скінченні групи 8
2.2.3 Циклічні групи 8
2.2.4 Теореми про циклічні групи 9
2.2.5 Підгрупи 9
2.2.6 Теорема Лагранжа 9
2.2.7 Циклічна підгрупа та Генератори 10
2.3 Дискретний логарифм 10
2.3.1 Формальне визначення дискретного логарифму 10
2.3.2 Приклади дискретного логарифму 10
2.3.3 Узагальнений дискретний логарифм (GDLP) 11
2.3.4 Приклад узагальненого дискретного логарифму 11
2.4 Безпека дискретного логарифму 11

3 Шифрування Ель-Гамаля (ElGamal) **13**
3.1 Основна ідея протоколу 13
3.2 Формальний опис схеми 13
3.2.1 Генерація ключів 13
3.2.2 Шифрування 14
3.2.3 Розшифрування 14
3.2.4 Коректність 15
3.3 Приклад 15
3.4 Безпека алгоритму Ель-Гамаля 15

4 Введення в еліптичні криві **17**
4.1 Мотивація та базові означення 17
4.2 Операції на еліптичних кривих 17
4.2.1 Додавання точок 17
4.2.2 Множення точки на скаляр (Point Multiplication) 18

5 Перехід від класичного ElGamal до ElGamal на основі еліптичних кривих **19**
5.1 Загальна ідея 19
5.2 Алгоритм ElGamal на еліптичній кривій 19
5.2.1 Генерація ключів 19
5.2.2 Шифрування 19
5.2.3 Розшифрування 20
5.3 Переваги та недоліки EC ElGamal 20

6 Графічне представлення результатів **21**

Висновки **22**

Список використаних джерел **22**

Вступ

У сучасному світі інформаційна безпека набуває все більшої значущості через постійне зростання обсягів переданої та збереженої інформації. Захист даних від несанкціонованого доступу, зміни чи видалення є ключовою задачею у сфері інформаційних технологій. Криптографія виступає основним інструментом для забезпечення конфіденційності, цілісності та автентичності інформації.

Криптографія має давню історію, починаючи ще з Стародавнього Єгипту, де використовувалися прості символічні шифри для передачі секретних повідомлень. У середньовіччі арабські вчені, такі як Аль-Хорезмі, зробили значний внесок у розвиток криптографії, створюючи складніші шифри та методи їх розшифровки.

У 20 столітті криптографія отримала новий імпульс завдяки появі комп'ютерів та сучасних методів математичного аналізу. Під час Другої світової війни розвиток криптографії досяг піку з розшифровкою німецьких шифрів «Енігма» Бертраном Расселом та іншими вченими. Ці події продемонстрували важливість криптографії для військових та державних потреб.

Після численних війн криптографія активно застосовується у цивільному секторі, особливо з розвитком електронної комунікації. Виникнення Інтернету та цифрових технологій підвищило потребу у надійних методах захисту інформації, що сприяло розвитку асиметричних криптографічних алгоритмів.

Метою цієї кваліфікаційної роботи є дослідження ефективності класичних алгоритмів криптографії на основі еліптичних кривих. Зокрема, буде проведено аналіз продуктивності певного класичного алгоритму та його варіації на основі еліптичних кривих, а також оцінено їх придатність для застосування у сучасних інформаційних системах.

Завданнями дослідження є:

- Ознайомлення з основними принципами та теорією криптографії, необхідними для даної кваліфікаційної роботи.
- Аналіз існуючих класичних криптографічних алгоритмів та їх варіацій, що використовують еліптичні криві.
- Проведення експериментального дослідження ефективності обраних алгоритмів.
- Порівняння отриманих результатів та формулювання висновків щодо їх практичної застосовності.

Розділ 1

Постановка задачі

1.1 Описова постановка

Нам задано класичний алгоритм асиметричної криптографії та його варіацію на основі еліптичних кривих. Метою дослідження є аналіз та пояснення процесу адаптації класичного алгоритму до версії на основі еліптичних кривих, а також оцінка їх ефективності, безпеки та практичної релевантності.

Зокрема, буде проведено порівняння обраного класичного алгоритму (Алгоритм 1) з його еквівалентною варіацією на основі еліптичних кривих (Алгоритм 2). Аналіз охоплюватиме наступні аспекти:

- **Процес адаптації:** Вивчення змін у математичній основі та алгоритмічній структурі при переході від Алгоритму 1 до Алгоритму 2.
- **Ефективність:** Оцінка швидкості виконання операцій шифрування та дешифрування, використання обчислювальних ресурсів та розміру ключів.
- **Безпека:** Аналіз стійкості алгоритмів до сучасних криптографічних атак, включаючи класичні та квантові атаки.
- **Практична релевантність:** Визначення придатності алгоритмів для застосування у різних умовах, таких як мобільні пристрої, вбудовані системи та великі інформаційні мережі.

Дослідження спрямоване на визначення переваг та недоліків класичних асиметричних алгоритмів у порівнянні з їх варіаціями на основі еліптичних кривих, а також надання рекомендацій щодо вибору оптимальних криптографічних методів для різних сфер застосування.

1.2 Математична модель

Рисунки 1.1 та 1.2 демонструють процес обміну зашифрованими повідомленнями між двома сторонами: відправником (Алісою) та отримувачем (Бобом) для класичного алгоритму та його еквіваленту на основі ECC відповідно.

- $k_{\text{pub}}^{(1)}$, $k_{\text{pub}}^{(2)}$: публічний ключ для класичного алгоритму та ECC відповідно.
- $k_{\text{pr}}^{(1)}$, $k_{\text{pr}}^{(2)}$: приватний ключ для класичного алгоритму та ECC відповідно.

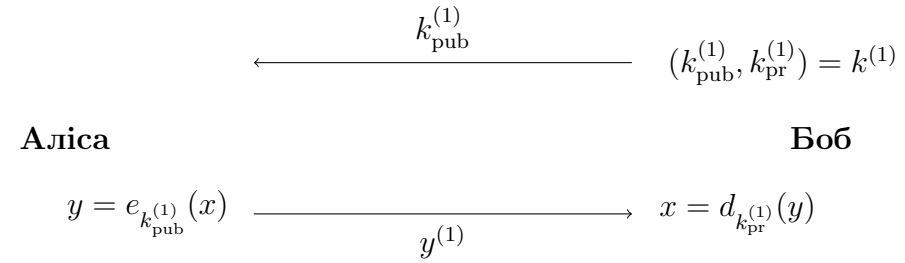

Рис. 1.1: Базовий протокол асиметричного шифрування на основі класичного алгоритму.

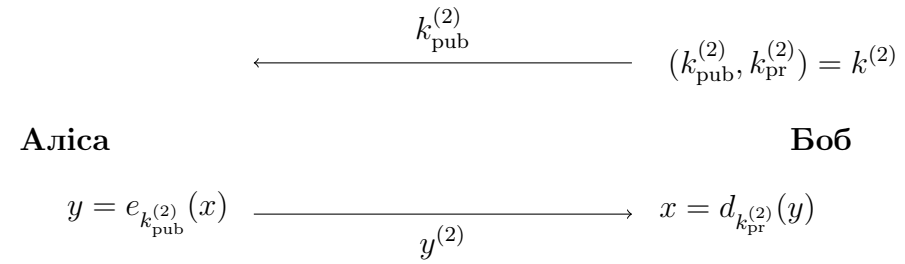

Рис. 1.2: Базовий протокол асиметричного шифрування на основі ECC.

- $k^{(1)}$, $k^{(2)}$: пара ключів (публічний і приватний) для класичного алгоритму та ECC відповідно.
- $e_{k^{(1)}_{\text{pub}}}$, $e_{k^{(2)}_{\text{pub}}}$: функція шифрування з використанням публічного ключа.
- $d_{k^{(1)}_{\text{pr}}}$, $d_{k^{(2)}_{\text{pr}}}$: функція дешифрування з використанням приватного ключа.
- x: вихідне повідомлення, яке потрібно зашифрувати.
- $y^{(1)}, y^{(2)}$: зашифроване повідомлення з використанням класичного алгоритму та ECC відповідно.

Розділ 2

Проблема дискретного логарифму

2.1 Огляд головних криптографічних схем

Сучасна криптографія спирається на три основні сімейства алгоритмів відкритого ключа, які мають практичну значущість і базуються на різних математичних проблемах. Кожне з цих сімейств забезпечує основні криптографічні функції, такі як встановлення ключів, цифрові підписи для нерепудійованості та шифрування даних. Нижче наведено короткий огляд цих трьох сімейств.

2.1.1 Схеми на основі факторизації цілих чисел

Схеми цього сімейства ґрунтуються на складності задачі факторизації великих цілих чисел. Найвідомішим представником є алгоритм RSA, запропонований у 1977 році. RSA широко використовується для шифрування, цифрових підписів та встановлення ключів. Ефективність RSA залежить від обчислювальної складності розкладу великих чисел на прості множники. При виборі належних параметрів, таких як довжина ключа, алгоритм забезпечує високий рівень безпеки.

2.1.2 Схеми на основі дискретного логарифму

Алгоритми цього сімейства базуються на задачі дискретного логарифму в скінченних полях. До найвідоміших представників належать:

- Протокол обміну ключами Діффі–Геллмана (DHKE),
- Шифрування Ель-Гамаля (ElGamal Encryption),
- Алгоритм цифрового підпису (Digital Signature Algorithm, DSA).

Ці алгоритми були запропоновані в середині 1970-х років і залишаються надійними за умови правильного вибору параметрів. В основі їх безпеки лежить складність обчислення дискретного логарифму — задачі, для якої не існує відомих ефективних алгоритмів розв'язання.

2.1.3 Схеми на основі еліптичних кривих

Ця група є узагальненням схем на основі дискретного логарифму. Алгоритми на основі еліптичних кривих (Elliptic Curve Cryptography, ECC) були запропоновані в

середині 1980-х років і мають перевагу у зменшенні розмірів ключів без втрати рівня безпеки. Найвідоміші приклади включають:

- Обмін ключами за допомогою еліптичних кривих (Elliptic Curve Diffie–Hellman, ECDH),
- Алгоритм цифрового підпису на основі еліптичних кривих (Elliptic Curve Digital Signature Algorithm, ECDSA).

Схеми ECC використовують коротші ключі порівняно з класичними алгоритмами, такими як RSA, забезпечуючи той самий рівень криптографічної безпеки. Це робить ECC привабливими для середовищ з обмеженими ресурсами, наприклад, мобільних пристроїв або вбудованих систем.

2.1.4 Інші схеми відкритого ключа

Окрім трьох основних сімейств, існують також інші схеми, такі як:

- Мультиваріативні квадратичні схеми (Multivariate Quadratic, MQ),
- Схеми на основі ґраток (Lattice-based schemes),
- Криптосистеми McEliece.

Проте ці схеми часто мають недостатню криптографічну зрілість або погані характеристики реалізації, такі як надмірно великі ключі. Інші схеми, наприклад, криптосистеми на основі гіпереліптичних кривих, є ефективними та безпечними, але поки що не набули широкого розповсюдження. Для більшості застосувань рекомендовано використовувати схеми з трьох основних сімейств.

2.2 Алгебраїчні структури в контексті дискретного логарифму

Для розуміння задачі дискретного логарифму важливо ввести базові алгебраїчні структури, які використовуються в криптографії. До них належать групи, циклічні групи та їхні підгрупи.

2.2.1 Групи

Визначення 2.1 (Група)**.** Групою G називається множина елементів разом з операцією $\circ$, яка комбінує два елементи G. Група задовольняє наступні властивості:

1. **Замкненість:** Для всіх $a, b \in G$ результат операції $a \circ b$ також належить G.
2. **Асоціативність:** $a \circ (b \circ c) = (a \circ b) \circ c$ для всіх $a, b, c \in G$.
3. **Нейтральний елемент:** Існує елемент $e \in G$, такий що $a \circ e = e \circ a = a$ для всіх $a \in G$.
4. **Обернений елемент:** Для кожного $a \in G$ існує $a^{-1} \in G$, такий що $a \circ a^{-1} = a^{-1} \circ a = e$.

Якщо додатково виконується **комутативність** ($a \circ b = b \circ a$), така група називається **абелевою**.

Приклад 2.1.
- Множина цілих чисел $\mathbb{Z}$ з операцією додавання утворює абелеву групу, де 0 є нейтральним елементом, а $-a$ є оберненим до a.
- Множина $\mathbb{Z}_n^*$ (всі числа, менші за n та взаємно прості з n) з операцією множення за модулем n утворює абелеву групу, якщо $n > 1$.

2.2.2 Скінченні групи

Визначення 2.2 (Скінченна група)**.** Група $(G, \circ)$ називається **скінченною**, якщо вона містить скінченну кількість елементів. Кількість елементів у групі називається її **порядком** і позначається $|G|$.

Приклад 2.2.
- Множина $\mathbb{Z}_n$ з операцією додавання за модулем n має порядок $|G| = n$.
- Для множини $\mathbb{Z}_9^* = \{1, 2, 4, 5, 7, 8\}$ порядок дорівнює $|\mathbb{Z}_9^*| = 6$.

2.2.3 Циклічні групи

Визначення 2.3 (Порядок елемента)**.** Порядком $\text{ord}(a)$ елемента a групи $(G, \circ)$ називається найменше додатне ціле число k, таке що $a^k = e$, де e — нейтральний елемент групи.

Приклад 2.3. Визначимо порядок елемента $a = 3$ у групі $\mathbb{Z}_{11}^*$:

$$\begin{aligned} a^1 &= 3 \\ a^2 &= 9 \mod 11 \\ a^3 &= 5 \mod 11 \\ a^4 &= 4 \mod 11 \\ a^5 &= 1 \mod 11 \end{aligned}$$

Отже, $\text{ord}(3) = 5$.

Визначення 2.4 (Циклічна група)**.** Група G називається **циклічною**, якщо існує елемент $\alpha \in G$ з порядком $\text{ord}(\alpha) = |G|$. Такий елемент α називається **генератором** або **примітивним елементом**, оскільки кожен елемент $g \in G$ можна записати як $g = \alpha^k$ для деякого k.

Приклад 2.4. Група $\mathbb{Z}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ є циклічною. Елемент 2 є генератором, оскільки степені $2^i \mod 11$ проходять усі елементи групи:

$$\{2, 4, 8, 5, 10, 9, 7, 3, 6, 1\}$$

Отже, $\text{ord}(2) = 10 = |\mathbb{Z}_{11}^*|$.

2.2.4 Теореми про циклічні групи

Теорема 2.1. *Якщо $|G| = n$, то порядок будь-якого елемента $a \in G$ ділить n.*

Теорема 2.2. *Якщо G є циклічною групою з порядком n, то кількість генераторів у G дорівнює $\varphi(n)$, де φ — функція Ейлера.*

Приклад 2.5. Для групи $\mathbb{Z}_{11}^*$ з порядком $|G| = 10$ кількість генераторів дорівнює $\varphi(10) = 4$. Ці генератори — елементи $\{2, 6, 7, 8\}$.

2.2.5 Підгрупи

Визначення 2.5 (Підгрупа)**.** Підгрупою $H \subset G$ називається підмножина H, яка сама є групою щодо операції $\circ$, визначеної в G.

Теорема 2.3 (Циклічна підгрупа)**.** *Нехай G є циклічною групою. Тоді кожен елемент $a \in G$ з порядком $ord(a) = s$ є генератором циклічної підгрупи з s елементами.*

Приклад 2.6. Розглянемо групу $\mathbb{Z}_{11}^*$. Елемент $a = 3$ має порядок $\text{ord}(3) = 5$, тому підгрупа, згенерована 3, містить елементи $\{1, 3, 4, 5, 9\}$. Перевіримо це за допомогою таблиці множення:

$\times$ mod 11	1	3	4	5	9
1	1	3	4	5	9
3	3	9	1	4	5
4	4	1	5	9	3
5	5	4	9	3	1
9	9	5	3	1	4

Табл. 2.1: Таблиця множення для підгрупи $H = \{1, 3, 4, 5, 9\}$ у групі $\mathbb{Z}_{11}^*$.

Отже, $H = \{1, 3, 4, 5, 9\}$ є підгрупою порядку 5.

2.2.6 Теорема Лагранжа

Теорема 2.4 (Теорема Лагранжа)**.** *Нехай H є підгрупою групи G. Тоді порядок підгрупи $|H|$ ділить порядок групи $|G|$.*

Приклад 2.7. Група $\mathbb{Z}_{11}^*$ має порядок $|G| = 10$. Відповідно до теореми Лагранжа, можливі порядки підгруп: $1, 2, 5, 10$. Підгрупи мають відповідні порядки:

- $H_1 = \{1\}$ з порядком 1,
- $H_2 = \{1, 10\}$ з порядком 2,
- $H_3 = \{1, 3, 4, 5, 9\}$ з порядком 5,
- $H_4 = \mathbb{Z}_{11}^*$ з порядком 10.

2.2.7 Циклічна підгрупа та Генератори

Теорема 2.5. *Нехай G є циклічною групою порядку n, і нехай α — її генератор. Тоді для кожного дільника k числа n існує рівно одна циклічна підгрупа H порядку k, яка генерується елементом $\beta = \alpha^{n/k}$. Підгрупа H складається з елементів $\{1, \beta, \beta^2, \ldots, \beta^{k-1}\}$.*

Приклад 2.8. Розглянемо групу $\mathbb{Z}_{11}^*$ з порядком $|G| = 10$ і генератором $\alpha = 8$. Хочемо знайти генератор підгрупи порядку $k = 2$:

$$\beta = \alpha^{10/2} = 8^5 \equiv 32768 \equiv 10 \mod 11.$$

Перевіримо, що $\beta = 10$ генерує підгрупу $H = \{1, 10\}$:

$$\beta^1 \equiv 10 \mod 11, \quad \beta^2 \equiv 100 \equiv 1 \mod 11.$$

Отже, $H = \{1, 10\}$ є підгрупою порядку 2, згенерованою елементом 10.

2.3 Дискретний логарифм

2.3.1 Формальне визначення дискретного логарифму

Дискретний логарифм у групі $\mathbb{Z}_p^*$, де p — просте число, є однією з ключових складових багатьох криптографічних протоколів. Його складність забезпечує безпеку криптосистем.

Визначення 2.6 (Дискретний логарифм (DLP) в $\mathbb{Z}_p^*$)**.** Дано скінченну циклічну групу $\mathbb{Z}_p^*$ порядку $p-1$, примітивний елемент $\alpha \in \mathbb{Z}_p^*$ та інший елемент $\beta \in \mathbb{Z}_p^*$. DLP полягає у визначенні цілого числа $1 \leq x \leq p-1$, такого що:

$$\alpha^x \equiv \beta \mod p$$

Таке число x називається дискретним логарифмом β до основи α і записується як:

$$x = \log_\alpha \beta \mod p.$$

Обчислення дискретних логарифмів за модулем простого числа є дуже складною задачею за умови достатньо великих параметрів. Оскільки експонентування $\alpha^x \equiv \beta \mod p$ є обчислювально легким, це утворює односторонню функцію.

2.3.2 Приклади дискретного логарифму

Приклад 2.9. Розглянемо дискретний логарифм у групі $\mathbb{Z}_{47}^*$, де $\alpha = 5$ є примітивним елементом. Для $\beta = 41$ задача дискретного логарифму полягає у визначенні позитивного цілого числа x, такого що

$$5^x \equiv 41 \mod 47$$

Використовуючи метод перебору, ми отримуємо розв'язок $x = 15$.

2.3.3 Узагальнений дискретний логарифм (GDLP)

Визначення 2.7 (Узагальнений дискретний логарифм (GDLP))**.** Дано скінченну циклічну групу G з операцією групи $\circ$ та порядком n. Розглядається примітивний елемент $\alpha \in G$ та інший елемент $\beta \in G$. Задача дискретного логарифму полягає у визначенні цілого числа x, де $1 \leq x \leq n$, такого що:

$$\beta = \alpha \circ \alpha \circ \ldots \circ \alpha = \alpha^x$$

(повторення α x разів).

Як і у випадку DLP у $\mathbb{Z}_p^*$, таке число x обов'язково існує, оскільки α є примітивним елементом, і таким чином кожен елемент групи G можна згенерувати шляхом повторного застосування операції групи до α.

2.3.4 Приклад узагальненого дискретного логарифму

Приклад 2.10 (Приклад 8.13)**.** Розглянемо адитивну групу цілих чисел за модулем простого числа. Нехай $p = 11$, група $G = (\mathbb{Z}_{11}, +)$ є скінченною циклічною групою з примітивним елементом $\alpha = 2$. Ось як α генерує групу:

i	1	2	3	4	5	6	7	8	9	10	11
$i\alpha$	2	4	6	8	10	1	3	5	7	9	0

Спробуємо вирішити DLP для елемента $\beta = 3$, тобто знайти x, таке що

$$x \cdot 2 \equiv 3 \mod 11$$

Щоб знайти x, інвертуємо 2 за модулем 11:

$$x \equiv 2^{-1} \cdot 3 \mod 11$$

За допомогою розширеного алгоритму Евкліда, $2^{-1} \equiv 6 \mod 11$, отже:

$$x \equiv 6 \cdot 3 \equiv 18 \equiv 7 \mod 11$$

Перевіримо:

$$7 \cdot 2 = 14 \equiv 3 \mod 11$$

Отже, $x = 7$.

Ми можемо узагальнити цей метод до будь-якої групи $(\mathbb{Z}_n, +)$ для довільного n та елементів $\alpha, \beta \in \mathbb{Z}_n$. Таким чином, узагальнений DLP є обчислювально легким у $\mathbb{Z}_n$, оскільки використовуються прості математичні операції, такі як множення та інверсія.

2.4 Безпека дискретного логарифму

У цьому розділі розглядаються основні методи атаки на задачу дискретного логарифму (DLP). Безпека багатьох асиметричних криптосистем базується на складності обчислення DLP у циклічних групах, тобто на визначенні x для даних α та β у групі G, таких що

$$\beta = \alpha^x$$

(повторення α x разів).

Складність вирішення DLP залежить від обраної групи та використовуваних алгоритмів атаки. Наразі не існує відомих алгоритмів, що вирішують DLP за поліноміальний час у загальному випадку, що робить цю задачу основою для безпеки багатьох криптографічних протоколів.

- **Алгоритми, залежні від розміру групи**:
  - **Метод перебору**: Найпростіший та найвитратніший метод з експоненціальною складністю $O(|G|)$. Він полягає в послідовному обчисленні степенів генератора α до тих пір, поки не буде знайдено x, таке що $\alpha^x = \beta$.
  - **Метод малий крок – великий крок (baby-step giant-step)**: Алгоритм з складністю $O(\sqrt{|G|})$, який використовує компроміс між часом виконання та використанням пам'яті. Він розділяє дискретний логарифм на дві частини та використовує таблиці для швидкого пошуку відповідей.
  - **Метод Ро Полларда (Pollard's Rho)**: Ймовірнісний алгоритм також зі складністю $O(\sqrt{|G|})$, але з меншими вимогами до пам'яті. Він використовує псевдовипадкову генерацію елементів групи та виявляє колізії для знаходження дискретного логарифму.
- **Алгоритми, залежні від розміру простих множників порядку групи**:
  - **Алгоритм Поліґа-Геллмана (Pohlig–Hellman)**: Використовує факторизацію порядку групи для розбиття задачі DLP на менші підзадачі в підгрупах з простими порядками. Це дозволяє значно зменшити обчислювальну складність, особливо коли порядок групи має малий найбільший простий дільник.
- **Метод Index Calculus**: Найпотужніший алгоритм для груп $\mathbb{Z}_p^*$ та $\mathrm{GF}(2^m)^*$ з субекспоненціальною складністю. Він базується на факторизації елементів групи через малу підмножину базових елементів, що дозволяє ефективно обчислювати дискретні логарифми. Проте цей метод не застосовується до груп на основі еліптичних кривих, що робить ECC більш безпечним вибором у цьому контексті.

Розділ 3

Шифрування Ель-Гамаля (ElGamal)

3.1 Основна ідея протоколу

Протокол Ель-Гамаля (ElGamal) — це одна з криптосистем відкритого ключа, яка забезпечує конфіденційність повідомлень на основі складності розв'язання задачі дискретного логарифму. Алгоритм був запропонований Тахером Ель-Гамалем у 1985 році й ґрунтується на протоколі обміну ключами Діффі–Геллмана.

Основні фази алгоритму Ель-Гамаля:

- **Генерація ключів (Set-up):** виконується отримувачем (Бобом), який бажає отримувати зашифровані повідомлення.
- **Шифрування (Encryption):** виконується відправником (Алісою) для кожного повідомлення.
- **Розшифрування (Decryption):** виконується Бобом для кожного отриманого шифротексту.

3.2 Формальний опис схеми

Нехай p — велике просте число, а α — примітивний елемент (генератор) у групі $\mathbb{Z}_p^*$ або (що часто зустрічається на практиці) у деякій підгрупі простого порядку цієї групи. Для унаочнення припустимо, що всі дії відбуваються за модулем p.

3.2.1 Генерація ключів

- **Крок 1.** Боб обирає велике просте число p. (Зазвичай p має довжину щонайменше 1024 біти.)
- **Крок 2.** Боб обирає примітивний елемент (генератор) $\alpha \in \mathbb{Z}_p^*$. Як правило, α генерує велику підгрупу простого порядку, щоб уникнути атак на малі підгрупи.
- **Крок 3.** Боб випадково обирає секретне число $d \in \{2, 3, \ldots, p-2\}$. Це його приватний ключ.
- **Крок 4.** Боб обчислює відкритий ключ $\beta = \alpha^d \bmod p$.

- **Крок 5.** Боб публікує свій відкритий ключ $\big(p, \alpha, \beta\big)$ у відкритому реєстрі або на своєму веб-сайті, а приватний ключ d зберігає у таємниці.

Таким чином, Bob : $(k_{\text{pub}}, k_{\text{pr}}) = \big(\{p, \alpha, \beta\}, d\big)$.

3.2.2 Шифрування

Припустімо, що Аліса хоче надіслати Бобу повідомлення x, де x належить $\mathbb{Z}_p^*$. Процес шифрування виглядає так:

- **Крок 1.** Аліса отримує публічний ключ Боба $\big(p, \alpha, \beta\big)$ (наприклад, з відкритої бази ключів).
- **Крок 2.** Аліса випадково обирає тимчасовий ("ефемерний") показник $i \in \{2, 3, \ldots, p-2\}$.
- **Крок 3.** Обчислює $k_E = \alpha^i \bmod p$. Це так званий *ефемерний ключ*, який передаватиметься у відкритому вигляді.
- **Крок 4.** Обчислює $k_M = \beta^i \bmod p$. Це *маскуючий ключ*, за допомогою якого «приховують» вихідний текст.
- **Крок 5.** Шифротекст формується у два елементи:

  $$y = x \cdot k_M \bmod p$$

  і пара $\big(k_E,\, y\big)$.
- **Крок 6.** Аліса надсилає Бобу пару $\big(k_E, y\big)$.

3.2.3 Розшифрування

Отримавши пару $\big(k_E, y\big)$, Боб може легко відновити повідомлення x, маючи свій приватний ключ d. Для цього він виконує:

- **Крок 1.** Обчислює $k_M = \big(k_E\big)^d \bmod p$.
- **Крок 2.** Знаходить обернений елемент до k_M за модулем p. Можна обчислити $k_M^{-1} \bmod p$ за допомогою розширеного алгоритму Евкліда або скористатись малою теоремою Ферма:

  $$k_M^{-1} \equiv k_M^{p-2} \bmod p$$

  (якщо k_M не дорівнює нулю за модулем p, тобто належить $\mathbb{Z}_p^*$).
- **Крок 3.** Відновлює текст:

  $$x = y \cdot k_M^{-1} \bmod p.$$

Таким чином, маючи d, Боб легко вираховує оригінальне x.

3.2.4 Коректність

Переконаємося, що описані вище кроки дійсно відтворюють вихідне повідомлення:

$$\begin{aligned} x &\equiv y \cdot k_M^{-1} \bmod p \\ &\equiv \left(x \cdot \beta^i\right) \cdot \left(\left(\alpha^i\right)^d\right)^{-1} \bmod p \\ &\equiv x \cdot \alpha^{d \cdot i} \cdot \alpha^{-d \cdot i} \bmod p \\ &\equiv x \bmod p. \end{aligned}$$

3.3 Приклад

Для ілюстрації розглянемо малий числовий приклад (нерекомендований для реальних застосувань через надто малий модуль):

Приклад 3.1. Нехай $p = 29$, $\alpha = 2$. Боб обирає $d = 12$ та обчислює $\beta = \alpha^d = 2^{12} \bmod 29 \equiv 7$. Отже, відкритий ключ Боба: $(p, \alpha, \beta) = (29, 2, 7)$, а приватний ключ $d = 12$.

Шифрування. Аліса хоче зашифрувати повідомлення $x = 26$.

- Випадково обирає $i = 5$.
- Обчислює $k_E = \alpha^i = 2^5 \bmod 29 \equiv 3$.
- Обчислює $k_M = \beta^i = 7^5 \bmod 29 \equiv 16$.
- Обчислює $y = x \cdot k_M \bmod 29 = 26 \cdot 16 \bmod 29 \equiv 10$.
- Надсилає Бобу пару $\left(k_E, y\right) = (3, 10)$.

Розшифрування. Боб отримує $(k_E, y) = (3, 10)$.

- Обчислює $k_M = (k_E)^d \bmod 29 = 3^{12} \bmod 29 \equiv 16$.
- Знаходить k_M^{-1} (наприклад, $16 \cdot 20 \equiv 1 \bmod 29$), тобто $k_M^{-1} = 20$.
- Відновлює $x = 10 \cdot 20 \bmod 29 \equiv 26$.

Таким чином, 26 успішно відновлено.

3.4 Безпека алгоритму Ель-Гамаля

Безпека Ель-Гамаля базується на складності задачі дискретного логарифму (DLP) та спорідненої з нею задачі Діффі–Геллмана (DHP). Основні можливі атаки:

- **Пасивна атака (прослуховування).** Супротивник, який перехоплює повідомлення, бачить лише p, α, β, ефемерний ключ k_E і шифротекст y. Йому доведеться розв'язувати DLP, щоб знайти або приватний ключ d (через $\beta = \alpha^d$), або випадкову величину i (через $k_E = \alpha^i$).
- **Атака з активною участю (man-in-the-middle).** Як і в багатьох протоколах з відкритими ключами, необхідно пересвідчитись, що саме ключ Боба використовується для шифрування. Для цього на практиці застосовують сертифікати та інфраструктуру відкритих ключів (PKI).

- **Повторне використання ефемерного показника i.** Якщо випадковий показник i колись повториться, то для двох різних повідомлень x_1 і x_2 буде згенеровано однаковий маскуючий ключ k_M, що робить криптосистему вразливою. Тому рекомендується ретельно стежити, щоб i не повторювався.

- **Атаки на невеликі підгрупи (small subgroup attack).** Для уникнення цієї проблеми часто обирають α, що генерує підгрупу простого порядку.

Важливо зазначити, що Ель-Гамаль є *ймовірнісною* криптосистемою: навіть якщо Аліса двічі шифрує однакове повідомлення x, з великою ймовірністю будуть отримані різні шифротексти завдяки рандомізації i.

Алгоритм Ель-Гамаля ілюструє потужність та гнучкість криптосистем, побудованих на основі дискретного логарифму. Його надійність обумовлена складністю DLP і тим, що швидке обчислення дискретних логарифмів у загальному випадку лишається невідомим. Проте належний вибір параметрів, уникнення малих підгруп, використання надійних випадкових величин для «ефемерних» показників та перевірка автентичності публічного ключа є вирішальними для безпеки та практичної реалізації криптосистеми Ель-Гамаля.

Розділ 4

Введення в еліптичні криві

4.1 Мотивація та базові означення

У сучасній криптографії еліптичні криві (англ. *Elliptic Curves*) займають особливе місце завдяки тому, що задача дискретного логарифму на еліптичних кривих (ECDLP) вважається складнішою порівняно з аналогічною задачею в класичних групах (наприклад, у $\mathbb{Z}_p^*$). Це дозволяє досягати високого рівня безпеки за відносно меншої довжини ключа, що є критично важливим для систем з обмеженими обчислювальними ресурсами.

Визначення 4.1 (Еліптична крива над простим полем)**.** Нехай p — велике просте число. *Еліптичною кривою* над полем $\mathbb{F}_p$ називають множину розв'язків рівняння

$$E: \quad y^2 \equiv x^3 + ax + b \mod p$$

разом із спеціальною *ненульовою* точкою $\mathcal{O}$ ("точкою на нескінченності"), де $a, b \in \mathbb{F}_p$ обираються таким чином, щоб $\operatorname{disc}(E) = 4a^3 + 27b^2 \not\equiv 0 \mod p$. Це забезпечує відсутність особливостей (singularities) на кривій.

Кожна точка $P \in E$ має координати (x, y) у полі $\mathbb{F}_p$. Разом із операцією додавання, яку визначено спеціальним чином, точки еліптичної кривої утворюють *циклічну групу*. Нейтральним елементом в цій групі є точка $\mathcal{O}$.

4.2 Операції на еліптичних кривих

4.2.1 Додавання точок

Нехай $P, Q \in E$. Визначимо $P + Q$ як «геометричну» операцію:

- Якщо $P \neq Q$, проводимо пряму, що з'єднує P і Q. Ця пряма перетинає криву E ще в одній точці R. Віддзеркаливши R відносно осі x, отримаємо $P + Q$.
- Якщо $P = Q$ (подвоєння точки $2P$), замість прямої «через P і Q» розглядаємо дотичну в P. Вона також перетинає E у деякій точці R, після чого відбиваємо R відносно осі x.
- Якщо одна з точок є $\mathcal{O}$, то $P + \mathcal{O} = P$.

У алгебраїчному вигляді (у полях $\mathbb{F}_p$) формули для $P + Q$ отримуються через операції додавання, множення, інвертування $\mod p$. Таким чином визначають **адитивну** групу точок кривої: $(E, +)$.

4.2.2 Множення точки на скаляр (Point Multiplication)

Операція dP ("скалярне множення" або "point multiplication") означає додавання точки P із собою d разів:

$$dP = \underbrace{P + P + \ldots + P}_{d \text{ разів}}.$$

Аналогією до α^d у класичних схемах $\mathbb{Z}_p^*$ є саме dP в групі точок E. Обчислювати dP наївним додаванням d разів неефективно, тому застосовують алгоритм *Double-and-Add (Подвоєння й Додавання)*, аналогічний до "square-and-multiply":

Визначення 4.2 (Double-and-Add алгоритм)**. Вхід:** Точка $P \in E$, ціле d зі степеневим поданням $d = \sum_{i=0}^{t} d_i 2^i$, де $d_i \in \{0, 1\}$. **Вихід:** $T = dP$.

1. Покласти $T = P$.
2. Для $i = t - 1$ до 0 виконати:
   (а) $T \leftarrow T + T$ (подвоєння)
   (б) Якщо $d_i = 1$, то $T \leftarrow T + P$.
3. Повернути T.

Таким чином, для t бітів скаляра d алгоритм виконує приблизно $1.5t$ операцій додавання/подвоєння точок.

Приклад 4.1. Нехай нам треба обчислити $26P$. У двійковому поданні $26 = (11010)_2$. Алгоритм переглядає біти числа зліва направо, кожного разу роблячи подвоєння, а якщо біт дорівнює 1 — ще й додавання P.

Розділ 5

Перехід від класичного ElGamal до ElGamal на основі еліптичних кривих

5.1 Загальна ідея

Подібно до того, як алгоритм Ель-Гамаля (ElGamal) у класичному варіанті $\mathbb{Z}_p^*$ спирається на складність дискретного логарифму, існує його версія для еліптичних кривих — EC ElGamal (Elliptic Curve ElGamal). Тут замість обчислення $\alpha^d \mod p$ та $\alpha^i \mod p$ використовують точки на еліптичній кривій та операцію dP. Безпека базується на задачі ECDLP (Elliptic Curve Discrete Logarithm Problem).

5.2 Алгоритм ElGamal на еліптичній кривій

Нехай задано:

- $(E, +)$ — еліптична крива над $\mathbb{F}_p$,
- P — точка-генератор великого порядку $n \approx p$.

Аналогія з класичним ElGamal виглядає так:

5.2.1 Генерація ключів

1. **Приватний ключ:** $d \in \{1, \ldots, n-1\}$.
2. **Публічний ключ:** $\beta = dP$.
3. Усі параметри (E, p, P, β) публікуються, а d зберігається у таємниці.

5.2.2 Шифрування

Нехай повідомлення M закодоване як точка $M \in E$ (існують різні методи «вкладення» бітових даних у точку кривої). Щоб зашифрувати M, відправник:

1. Випадково обирає ефемерний показник $i \in \{1, \ldots, n-1\}$.
2. Обчислює ефемерний ключ $k_E = iP$.
3. Обчислює маскуючий ключ $k_M = i\beta = i(dP) = (id)P$.
4. Формує шифротекст $C = \big(k_E,\ M + k_M\big)$.

5.2.3 Розшифрування

Одержувач, який знає свій приватний ключ d, отримавши пару $(k_E,\ M + k_M)$:

1. Обчислює $d \cdot k_E = d(iP) = (di)P = k_M$.
2. Відновлює повідомлення: $M = (M + k_M) - k_M$.

5.3 Переваги та недоліки EC ElGamal

- **Переваги:**
  - Менший розмір ключів порівняно з класичними схемами на $\mathbb{Z}_p^*$ (RSA, DSA/ElGamal тощо) при однаковому рівні безпеки.
  - Вищий темп обчислень (шифрування/розшифрування) у середовищах з обмеженими ресурсами (смарт-картки, мобільні пристрої тощо).
  - Стійкість до потужних методів індекс-числення, які ефективно застосовуються у звичайній групі $\mathbb{Z}_p^*$.
- **Недоліки:**
  - Складніша реалізація: необхідне ретельне тестування кривих, щоб уникнути «слабких сімейств».
  - Складність (і нерідко патентні обмеження) у реалізації «вкладення» довільних бітових повідомлень у точки E.
  - Безпека суттєво залежить від вибору правильної кривої, її параметрів і порядку підгрупи.

Розділ 6

Графічне представлення результатів

У цьому розділі представлено графік, що ілюструє залежність часу шифрування (у мілісекундах) від розміру повідомлення (у мегабайтах) для класичного алгоритму ElGamal та його варіанту на основі еліптичних кривих (EC ElGamal). Графік створено за допомогою зовнішньої програми, а тут він вставлений як зображення.

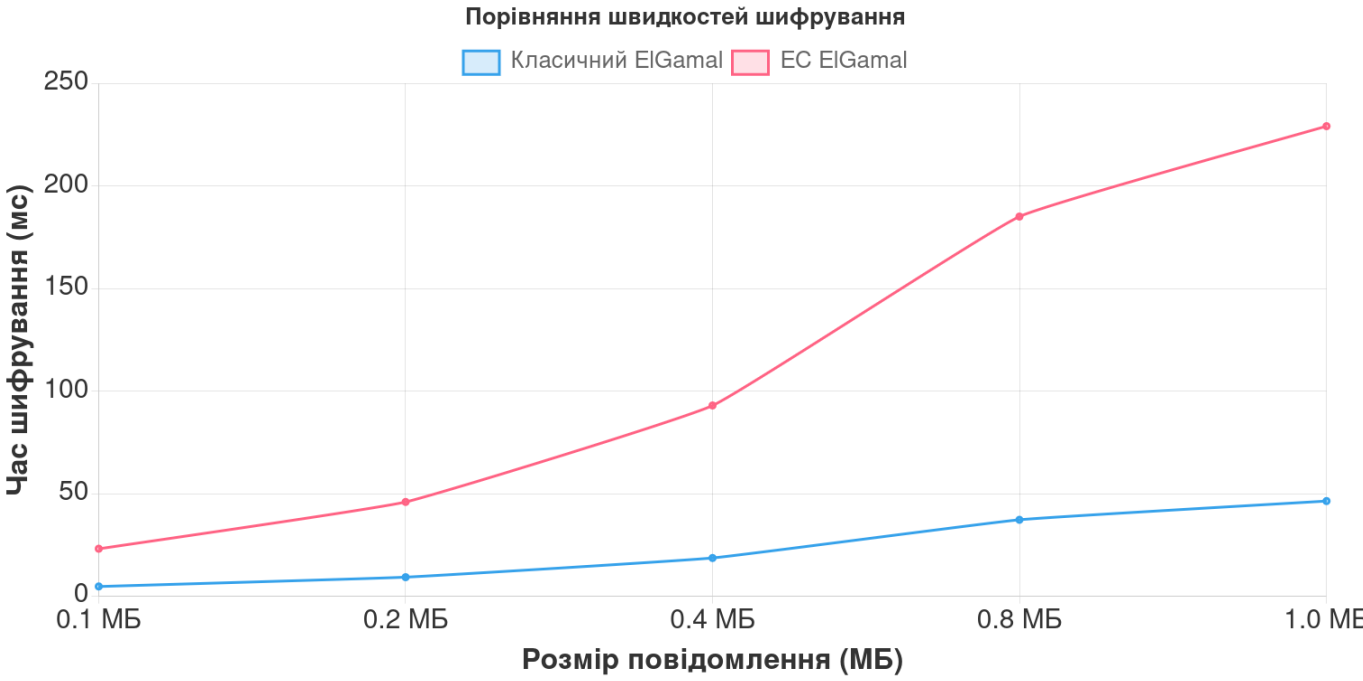

Рис. 6.1: Залежність часу шифрування від розміру повідомлення для класичного ElGamal та EC ElGamal.

Для вставки зображення використовується команда `\includegraphics` з пакету `graphicx`.

Висновки

У даній роботі було проведено комплексний аналіз ефективності класичних алгоритмів криптографії та їх адаптації на основі еліптичних кривих. Розглянуто математичні основи криптографічних схем, зокрема задачі дискретного логарифму та її узагальнення для еліптичних кривих (ECDLP). Аналіз протоколів ElGamal та EC ElGamal дозволив зробити наступні висновки:

- Адаптація класичного алгоритму ElGamal до еліптичних кривих дозволяє забезпечити еквівалентний рівень безпеки за значно меншого розміру ключа, що є критично важливим для сучасних систем з обмеженими обчислювальними ресурсами.
- Використання еліптичних кривих забезпечує вищу ефективність операцій шифрування та розшифрування, що було підтверджено експериментальними результатами, зображеними на графіку.
- Незважаючи на переваги, реалізація криптосистем на основі еліптичних кривих вимагає особливої уваги до вибору параметрів та ретельного тестування обраних кривих для уникнення потенційних вразливостей (наприклад, атак на малі підгрупи).
- Практична релевантність досліджуваних підходів підтверджується їх здатністю забезпечувати конфіденційність, цілісність та автентичність інформації, що робить їх перспективними для застосування у широкому спектрі сучасних інформаційних систем.

Отже, результати дослідження свідчать про доцільність впровадження еліптичних кривих у криптографічні схеми, що дозволяє оптимізувати використання ресурсів без компромісу щодо рівня безпеки.

Бібліографія

[1] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[2] N. Koblitz, *Elliptic Curve Cryptosystems*, Mathematics of Computation, vol. 48, no. 177, pp. 203–209, 1987.

[3] V. S. Miller, *Use of Elliptic Curves in Cryptography*, Advances in Cryptology — CRYPTO '85 Proceedings, pp. 417–426, 1986.

[4] T. ElGamal, *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, IEEE Transactions on Information Theory, vol. 31, no. 4, pp. 469–472, 1985.