

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

Кафедра математичних методів захисту інформації

Звіт  
з комп'ютерного практикуму  
«Реалізація основних асиметричних криптосистем»  
з кредитного модуля  
«Сучасні алгебраїчні криптосистеми»  
Варіант 2В

Виконала студентка  
Групи ФІ-52МН  
Остаповець Олеся

Київ 2025

# 1. Теоретичні відомості

Цифровий підпис – це криптографічний механізм, який забезпечує:

- автентичність – підпис створено власником секретного ключа;
- цілісність – будь-яка зміна повідомлення робить підпис недійсним;
- незаперечність – підписант не може правдоподібно заперечити факт підписання за умови збереження секретного ключа.

На відміну від шифрування, підпис не приховує зміст повідомлення, а підтверджує його походження та незмінність.

**ECDSA (Elliptic Curve Digital Signature Algorithm)** — алгоритм цифрового підпису на основі еліптичних кривих. Його стійкість базується на складності задачі дискретного логарифмування в групі точок еліптичної кривої. У порівнянні з RSA, ECDSA забезпечує аналогічний рівень безпеки при суттєво менших розмірах ключів наприклад, P-256 часто порівнюють з RSA-3072 за рівнем стійкості.

Генерація ключів:

1. Обирається випадкове число  $d \in [1, n - 1]$  – секретний ключ.
2. Обчислюється відкритий ключ  $Q = dG$  (точка на кривій)

Формування підпису.

Для повідомлення  $m$ :

1. Обчислити  $e = H(m)$ , – криптографічна хеш-функція (у реалізації використано SHA-256)
2. Згенерувати випадкове одноразове число  $k \in [1, n - 1]$  (nonce).
3. Обчислити точку  $R = kG$ , взяти  $r = R_x \bmod n$  (координата  $x$  по модулю  $n$ ). Якщо  $r = 0$ , вибрать інше  $k$ .
4. Обчислити  $s = k^{-1}(e + dr) \bmod n$ . Якщо  $s = 0$ , вибрать інше  $k$ .
5. Підпис – пара  $(r, s)$ .

Значення  $k$  має бути випадковим і ніколи не повторюватися. Повторне або передбачуване  $k$  може розкрити секретний ключ  $d$ .

Перевірка підпису

Для відкритого ключа  $Q$ , повідомлення  $m$  і підпису  $(r, s)$ :

1. Перевірити, що  $r$  і  $s$  належать  $[1, n - 1]$ . Якщо ні — підпис недійсний.

2. Обчислити  $e = H(m)$ .
3. Обчислити  $w = s^{-1} \bmod n$ .
4. Обчислити  $u_1 = ew \bmod n$ ,  $u_2 = rw \bmod n$ .
5. Обчислити точку  $X = u_1 G + u_2 Q$ .
6. Підпис коректний, якщо  $r \equiv X_x \bmod n$ .

## 2. Практична частина

Для кожної кривої (P-256, P-384) виконано:

1. Генерацію ключової пари ECDSA (приватний та публічний ключ).
2. Формування підпису для повідомлення:  $\text{sig} = \text{Sign}(\text{priv}, \text{SHA256}(\text{msg}))$ .
3. Перевірку підпису:  $\text{Verify}(\text{pub}, \text{SHA256}(\text{msg}), \text{sig})$ .

Результат контрольного прикладу:

- `demo_verify_ok` = True для всіх повідомлень та для обох кривих, тобто підпис коректно перевіряється, реалізація працює правильно.

Негативні тести:

Щоб показати властивість цілісності, виконано два негативні сценарії:

- Змінене повідомлення: інвертувався останній байт повідомлення, перевірка має провалитися.
- Змінений підпис: інвертувався останній біт підпису, перевірка має провалитися.

Отримано:

- `negative_test_modified_message_ok` = False
- `negative_test_modified_signature_ok` = False

для всіх повідомлень та для обох кривих. Це означає, що будь-яка модифікація повідомлення або підпису робить підпис недійсним, і механізм підпису виконує свою функцію контролю цілісності.

## Результати вимірювань

Крива	Public key (DER), байт	Private key (DER), байт	Підпис, байт
P-256	91	138	64
P-384	120	185	96

Спостерігається очікуване збільшення розміру підпису та ключів при переході з P-256 на P-384.

## Час виконання операцій (мкс/операцію)

Крива	Довжина повідомлення (байт)	Avg sign (μs)	Avg verify (μs)
P-256	2	280.3085	923.1420
P-256	68	299.15095	933.29675
P-256	4096	313.20185	974.3140
P-384	2	670.41615	2321.6169
P-384	68	632.39540	3434.60810
P-384	4096	858.66960	3142.77620

## Висновки

У ході виконання лабораторної роботи було реалізовано алгоритм цифрового підпису ECDSA із використанням хеш-функції SHA-256 та еліптичних кривих P-256 і P-384, а також проведено контрольний приклад підпису і перевірки повідомлень різної довжини. Перевірка коректного підпису в усіх випадках завершувалась успішно, а негативні тести зі зміненим повідомленням і зміненим підписом підтвердили забезпечення цілісності та неможливість непомітної модифікації даних. За результатами вимірювань встановлено, що переход з P-256 на P-384 збільшує обчислювальні витрати та розміри вихідних даних: підпис зрос з 64 до 96 байт, а середній час виконання операцій збільшився приблизно у 2.42 разів для підпису ( $\approx 297.55$  мкс  $\rightarrow \approx 720.49$  мкс) та у 3.14× для перевірки ( $\approx 943.58$  мкс  $\rightarrow \approx 2966.33$  мкс). Таким чином, P-256 є більш продуктивним варіантом для практичного застосування за обмежених ресурсів, тоді як P-384 забезпечує вищий рівень криптографічної стійкості ціною збільшення часу обробки та розмірів ключів/підпису.