АНАЛІЗ БЕЗПЕКИ СМАРТ-КОНТРАКТІВ НА ОСНОВІ ECDSA

Мокрій В.С., В'юхін Д.О.
Харківський національний університет радіоелектроніки, Харків, Україна

Цифровий підпис - ключовий механізм забезпечення автентичності операцій і цілісності даних в блокчейн-системах. У смарт-контрактах надійність підпису має особливе значення, оскільки помилки при роботі з ключами можуть призвести до несанкціонованого виконання операцій. За даними CertiK, Ethereum зафіксував 310 інцидентів у 2025 році, завдавши збитків приблизно на 1, 70 мільярда доларів. США [1]. Наведені статистичні дані демонструють масштаб практичних ризиків безпеки в екосистемі Ethereum.

Метою даної роботи є дослідження безпеки смарт-контрактів при використанні ECDSA. Враховуються властивості алгоритму, практичні ризики при створенні та перевірці підпису, варіанти зменшення ризиків. Особлива увага приділяється повторному використанню підпису та формуванню підписаних даних.

Мережа Ethereum використовує ECDSA на кривій secp256k1 для зовнішніх рахунків. Безпека алгоритму базується на складності задачі дискретного логарифма. Підпис генерується за допомогою закритого ключа та одноразового параметра nonce, і перевірка визначає, що він відповідає відкритому ключу. Критичною практичною загрозою для ECDSA є неправильне формування nonce. Його повторюване або передбачуване значення може створити умови для відновлення закритого ключа. Тому реалізація підпису повинна забезпечувати належне формування разових параметрів. Компрометація закритого ключа також дозволяє зловмиснику підписувати транзакції від імені власника [2].

Безпека ECDSA в смарт-контрактах також залежить від способу використання підпису. Договір повинен перевірити підписанта, затверджену операцію та факт попереднього використання підпису. Відтворення підпису - це повторне використання раніше створеного правильного підпису для повторного виконання операції. Щоб зменшити ризик, до підписуваних даних можна включити унікальний nonce операції, ідентифікатор мережі та адресу контракту [4].

Стійкість ECDSA до атак у смарт-контрактах визначається як криптографічною складністю основної математичної задачі, так і правильною реалізацією механізмів формування та перевірки цифрових підписів. Для кривої secp256k1 класична криптографічна стійкість пов'язана зі складністю задачі дискретного логарифмування на еліптичній кривій. На практиці, однак, порушення безпеки часто виникають не через прямий криптоаналіз ECDSA, а скоріше через помилки у використанні алгоритму. Критичними моментами є неправильна генерація одноразового параметра nonce, компрометація закритих ключів і повторне використання попередньо згенерованого правильного підпису (signature replay) [3]. Повторне використання або

передбачуваність nonce може створити умови для математичного відновлення закритого ключа, коли є достатньо підписів, тоді як компрометація закритого ключа фактично усуває необхідність криптоаналітичного обходу ECDSA, дозволяючи зловмиснику самостійно формувати дійсні підписи. Особливою загрозою для смарт-контрактів є повторна атака: навіть криптографічно коректний підпис може бути повторно використаний, якщо протокол не встановлює унікальний зв'язок між підписом і конкретною операцією, її параметрами і контекстом виконання. У такому випадку зловмиснику не потрібно підробляти підпис - достатньо повторно передати вже валідний підпис для виконання тієї самої або некоректно прив’язаної операції. Тому оцінювання стійкості ECDSA у смарт-контрактах повинно враховувати не лише математичну складність підроблення підпису, а й захищеність повного процесу його формування, передачі та верифікації.

Таблиця 1 – Основні загрози безпеці ECDSA у смарт-контрактах

Загроза	**Причина**	**Можливий наслідок**	**Основний захист**
Некоректний nonce	Повторне або передбачуване значення	Відновлення закритого ключа	Безпечне формування nonce
Компрометація ключа	Витік приватного ключа	Несанкціоновані підписи	Захист і безпечне зберігання ключа
Signature replay	Повторне використання коректного підпису	Повторне виконання операції	Nonce операції та перевірка контексту

Криптографічна стійкість ECDSA і стійкість смарт-контракту проти атак не є тотожними поняттями. Навіть якщо практична атака, заснована на математиці ECDSA, неможлива, безпека системи на рівні протоколу взаємодії смарт-контрактів може бути порушена. Особливе значення має перевірка унікальності транзакції: в договорі має бути зазначено, чи вже використовувався конкретний підпис або відповідна транзакція, а також перевіряти належність підписаних даних саме цьому контракту, мережі та операції, для яких вони були сформовані. Такий підхід зменшує ризик перенесення або повторного виконання дійсного підпису в інший контекст.

Для захисту підписаних даних у середовищі Ethereum застосовується EIP-712, який визначає стандартизований метод хешування та підпису типізованих структурованих даних [4]. Його використання разом з ECDSA дозволяє більш чітко визначити контекст, в якому підпис дійсний, особливо пов'язати підписані дані з мережевим ідентифікатором і адресою конкретного смарт-контракту. EIP-712 має важливе значення для зв’язку підписаних даних із визначеним контекстом і зменшення ризику їх повторного використання в іншому контексті. При цьому EIP-712 не змінює математичної основи ECDSA і не підвищує опір самої еліптичної кривої криптоаналітичним атакам. Його роль полягає у зниженні ризиків, пов'язаних із неправильним створенням та

інтерпретацією підписаних даних на протокольному рівні. Досліджені загрози показують, що стійкість ECDSA у смарт-контрактах має оцінюватися на двох взаємопов'язаних рівнях. На криптографічному рівні вирішальним є захист математичних основ алгоритму та правильне формування одноразового параметра nonce. На рівні смарт-контракту важливо забезпечити унікальне призначення підпису: підписані дані повинні відповідати конкретній операції, мережі та контракту. Саме порушення такого з'єднання може дозволити повторне використання правильного підпису, навіть якщо сам алгоритм ECDSA не був порушений.

Таким чином, відтворення підпису показує, що перевірка криптографічної правильності підпису сама по собі не гарантує безпечного виконання операції. Для протидії такій загрозі смарт-контракт повинен не тільки перевірити дійсність підпису, але й унікальність транзакції та узгодженість підписаних даних із зазначеним контекстом. Це дозволяє відокремити безпеку самого механізму цифрового підпису від безпеки його застосування в смарт-контракті.

Для практичного використання ECDSA найбільш критичними є атаки, які пов'язані не з прямим зломом математичної основи алгоритму, а з помилками в його застосуванні. Повторюване значення nonce несе в собі ризик компрометації закритого ключа, а відсутність прив'язки підпису до конкретної операції несе в собі ризик відтворення підпису. Використання nonce операції та структурованого підпису з визначеним доменом дозволяє врахувати ці загрози під час розробки смарт-контракту [5, 6].

Таким чином, стійкість смарт-контрактів на основі ECDSA до атак визначається поєднанням криптографічної стійкості алгоритму та безпечної реалізації механізмів підпису. Критичні умови включають правильне формування nonce, захист закритого ключа, забезпечення унікальності операцій і перевірку контексту підпису. Поєднання цих заходів дозволяє знизити ризик компрометації ключів і несанкціонованого повторного виконання операцій.

Список літератури

1. CertiK. Hack3d: The Web3 Security Report 2025. 2025.
2. National Institute of Standards and Technology. Digital Signature Standard (DSS). FIPS 186-5. 2023.
3. Peschanenko V., Rud V. Overview of Vulnerabilities in Smart Contracts Written in Solidity. Journal of Information Technologies in Education (ITE). 2024. № 56. P. 58–76.
4. EIP-712: Typed structured data hashing and signing. Ethereum Improvement Proposals.
5. Pornin, T. (2013) Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA). RFC 6979. Available at: https://doi.org/10.17487/RFC6979.
6. Bloemen, R., Logvinov, L. and Evans, J. (2017) Typed structured data hashing and signing. Ethereum Improvement Proposal 712 (EIP-712). URL: https://eips.ethereum.org/EIPS/eip-712.