МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

ЩИПАНСЬКИЙ РОМАН ВОЛОДИМИРОВИЧ

Метод захисту криптовалютних транзакцій / Cryptocurrency Transaction Protection Method

спеціальність: 125 – Кібербезпека та захист інформації

освітньо-професійна програма – Кібербезпека

Кваліфікаційна робота

Виконав студент групи КБм -21
Р.В. Щипанський

Науковий керівник
д.т.н., професор М.М.Касянчук

Кваліфікаційну роботу
допущено до захисту:

«____» _____ ___2025 р.

Завідувач кафедри
_____ В.В.Яцків

ТЕРНОПІЛЬ - 2025

Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки
Освітній ступінь «магістр»
спеціальність: 125 - Кібербезпека та захист інформації
освітньо-професійна програма –Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ В.В.Яцків
____" _____ 2024 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ЩИПАНСЬКОМУ Роману Володимировичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Метод захисту криптовалютних транзакцій / Cryptocurrency Transaction Protection Method
керівник роботи д.т.н., професор М.М.Касянчук
затверджені наказом по університету від 20 грудня 2024 року № 938

2. Строк подання студентом закінченої кваліфікаційної роботи 5 грудня 2025року.
3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.
4. Основні питання, які потрібно розробити:
– проаналізувати архітектуру блокчейн-технологій та систематизувати загрози безпеки;
– дослідити криптографічні методи захисту та алгоритми консенсусу;
– розробити алгоритми машинного навчання для детекції кіберзагроз;
– створити масштабовану архітектуру системи моніторингу;
– здійснити практичну реалізацію та тестування системи.
5. Перелік графічного матеріалу у роботі.
– ER-діаграма моделі виявлення аномалій у блокчейн-транзакціях;
– Багаторівнева архітектура системи моніторингу блокчейн-загроз;
– Архітектура Kubernetes кластера системи захисту криптовалютних транзакцій;
– Алгоритм адаптивної корекції порогу виявлення аномалій з механізмом зворотного зв'язку.

6. Консультанти розділів кваліфікаційної роботи

	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Теоретичні основи безпеки блокчейн-технологій та криптовалютних транзакцій	12.2024 р. – 03.2025 р.	
2	Криптографічні методи та алгоритми захисту блокчейн-транзакцій	03.2025 р. – 06.2025 р.	
3	Система моніторингу та виявлення аномалій у криптовалютних мережах	06.2025 р. – 11.2025 р.	

Студент _____ Р.В.Щипанський
(підпис)

Керівник роботи _____ д.т.н., проф. Касянчук М.М
(підпис)

АНОТАЦІЯ

Щипанський Р.В. Метод захисту криптовалютних транзакцій. – Рукопис.

Дослідження на здобуття освітнього ступеня «магістр» за спеціальністю 125 «Кібербезпека та захист інформації», освітньо-професійна програма «Кібербезпека». – Західноукраїнський національний університет, Тернопіль, 2025.

У роботі вирішується актуальна задача підвищення ефективності захисту криптовалютних транзакцій шляхом розробки комплексної системи моніторингу та виявлення кіберзагроз на основі інтеграції криптографічних методів та алгоритмів машинного навчання. Запропонована система поєднує криптографічні механізми протокольного рівня з інтелектуальним аналізом поведінкових патернів для виявлення фішингових атак, Ponzi-схем та відмивання коштів.

Розроблено багатошарову архітектуру з алгоритмом адаптивної корекції порогів та механізмом зворотного зв'язку, що забезпечує автоматичну адаптацію до еволюції загроз. Експериментальне дослідження на реальних блокчейн-датасетах з понад 1 мільйоном транзакцій підтвердило високу ефективність: точність детекції 94,7%, F1-Score 0,943, що на 15-20% краще за існуючі рішення.

Практично реалізована система підтримує пропускну здатність 73 000 транзакцій на секунду, забезпечує наскрізну затримку обробки 47-85 мілісекунд та включає інтеграцію з Kubernetes для автоматичного масштабування та самовідновлення при збоях.

Ключові слова: БЛОКЧЕЙН-БЕЗПЕКА, КРИПТОВАЛЮТНІ ТРАНЗАКЦІЇ, ВИЯВЛЕННЯ АНОМАЛІЙ, МАШИННЕ НАВЧАННЯ, КІБЕРБЕЗПЕКА, СИСТЕМА МОНІТОРИНГУ.

ABSTRACT

Shchypanskyi R.V. Cryptocurrency Transaction Protection Method. – Manuscript.

Research for obtaining the educational degree «Master» in specialty 125 «Cybersecurity and Information Protection», educational and professional program «Cybersecurity». – West Ukrainian National University, Ternopil, 2025.

The work addresses the urgent task of improving the effectiveness of cryptocurrency transaction protection through the development of a comprehensive monitoring and cyber threat detection system based on the integration of cryptographic methods and machine learning algorithms. The proposed system combines protocol-level cryptographic mechanisms with intelligent behavioral pattern analysis for detecting phishing attacks, Ponzi schemes, and money laundering.

A multi-layered architecture with adaptive threshold correction algorithm and feedback mechanism has been developed, ensuring automatic adaptation to threat evolution. Experimental research on real blockchain datasets with over 1 million transactions confirmed high efficiency: detection accuracy of 94.7%, F1-Score of 0.943, which is 15-20% better than existing solutions.

The practically implemented system supports throughput of 73,000 transactions per second, provides end-to-end processing latency of 47-85 milliseconds, and includes integration with Kubernetes for automatic scaling and self-recovery in case of failures.

Key words: BLOCKCHAIN SECURITY, CRYPTOCURRENCY TRANSACTIONS, ANOMALY DETECTION, MACHINE LEARNING, CYBERSECURITY, MONITORING SYSTEM.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AES – Advanced Encryption Standard (розширений стандарт шифрування);

API – Application Programming Interface (програмний інтерфейс додатку);

BFT – Byzantine Fault Tolerance (стійкість до візантійських відмов);

CNN – Convolutional Neural Network (згорткова нейронна мережа);

DApp – Decentralized Application (децентралізований додаток);

DDoS – Distributed Denial of Service (розподілена атака типу «відмова в обслуговуванні»);

DPoS – Delegated Proof of Stake (делегований доказ частки);

ECDLP – Elliptic Curve Discrete Logarithm Problem (задача дискретного логарифмування на еліптичних кривих);

ECDSA – Elliptic Curve Digital Signature Algorithm (алгоритм цифрового підпису на еліптичних кривих);

GPU – Graphics Processing Unit (графічний процесор);

LSTM – Long Short-Term Memory (довга короткострокова пам'ять);

ML – Machine Learning (машинне навчання);

MPC – Multi-Party Computation (багатосторонні обчислення);

NFT – Non-Fungible Token (невзаємозамінний токен);

PoS – Proof of Stake (доказ частки) PoW – Proof of Work (доказ роботи);

REST – Representational State Transfer (передача репрезентативного стану);

RPC – Remote Procedure Call (віддалений виклик процедур);

SHA – Secure Hash Algorithm (алгоритм безпечного хешування);

SIEM – Security Information and Event Management (управління інформацією та подіями безпеки);

ZKP – Zero-Knowledge Proof (доказ з нульовим розголошенням);

zk-SNARK – Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (стислий неінтерактивний аргумент знань з нульовим розголошенням);

zk-STARK – Zero-Knowledge Scalable Transparent Argument of Knowledge (масштабований прозорий аргумент знань з нульовим розголошенням).

ВСТУП

Актуальність роботи. Стрімкий розвиток блокчейн-технологій супроводжується зростанням кіберзагроз. За даними «Chainalysis» 2024 року, втрати від атак на блокчейн-платформи перевищили 3,8 мільярда доларів США, що на 47% більше порівняно з попереднім роком. Традиційні методи забезпечення безпеки, що базуються виключно на криптографічних механізмах, недостатньо ефективні проти сучасних загроз, які експлуатують людський фактор та помилки розробників. Інтеграція алгоритмів машинного навчання відкриває можливості для створення адаптивних систем захисту.

Мета і завдання дослідження. Мета роботи - розробка комплексної системи моніторингу та захисту криптовалютних транзакцій на основі інтеграції криптографічних методів та алгоритмів машинного навчання для виявлення кіберзагроз у реальному часі.

Завдання дослідження:

1) проаналізувати архітектуру блокчейн-технологій та систематизувати загрози безпеки;
2) дослідити криптографічні методи захисту та алгоритми консенсусу;
3) розробити алгоритми машинного навчання для детекції кіберзагроз;
4) створити масштабовану архітектуру системи моніторингу;
5) здійснити практичну реалізацію та тестування системи.

Об'єкт дослідження - процеси забезпечення безпеки блокчейн-технологій та криптовалютних транзакцій.

Предмет дослідження - методи виявлення та класифікації кіберзагроз у блокчейн-мережах.

Методи дослідження включають теоретичний аналіз сучасних криптографічних алгоритмів та механізмів консенсусу блокчейн-систем, експериментальне дослідження алгоритмів машинного навчання (Isolation Forest, XGBoost, LSTM) для виявлення аномалій у криптовалютних транзакціях,

а також практичне моделювання та навантажувальне тестування розробленої системи на реальних датасетах з понад 1 мільйоном транзакцій.

Науковою новизною дослідження є запропонований комплексне рішення для захисту інформаційних систем, яке базується на інтегрованому підході, що органічно поєднує криптографічні методи з технологіями машинного навчання. Ключовими досягненнями роботи стали удосконалення алгоритму виявлення аномалій завдяки впровадженню механізму адаптивної корекції порогів, що підвищує точність детекції загроз.

Практичне значення отриманих результатів. Розроблена система моніторингу та захисту криптовалютних транзакцій має безпосереднє практичне застосування для криптовалютних бірж, платформ децентралізованих фінансів, провайдерів гаманців, регуляторних органів та корпоративних блокчейн-рішень. Система забезпечує автоматизоване виявлення фішингових атак з точністю 94,7%, детекцію Ponzi-схем та схем відмивання коштів, моніторинг підозрілих транзакцій у режимі реального часу та генерацію сповіщень для оперативного реагування служб безпеки.

Результати дослідження. Розроблені алгоритми та архітектурні рішення можуть бути адаптовані для різних блокчейн-платформ (Bitcoin, Ethereum, Binance Smart Chain та інших) з мінімальними модифікаціями завдяки модульній структурі системи. Результати дослідження також мають освітню цінність та можуть використовуватися у навчальному процесі при підготовці фахівців з кібербезпеки та блокчейн-технологій.

Публікакації та апробації.

1. Щипанський Р., Іваніцький Р. Кібербезпека в контексті сучасних конфліктів. Матеріали XIV Міжнародної науково-технічної конференції «ITSec-2025: Безпека інформаційних технологій». Тернопіль, 22–24 травня 2025 р. с. 222-224

2. Щипанський Р., Бабала Л. Аналіз безпеки блокчейн-технологій та розробка методів захисту криптовалютних транзакцій. Матеріали науково-практичного симпозіуму «Захист інформації'2025». Тернопіль, 2025. с. 110-112

1 ТЕОРЕТИЧНІ ОСНОВИ БЕЗПЕКИ БЛОКЧЕЙН-ТЕХНОЛОГІЙ ТА КРИПТОВАЛЮТНИХ ТРАНЗАКЦІЙ

1.1 Фундаментальні принципи та архітектура блокчейн-технологій

Блокчейн – це розподілена база даних, що складається з блоків, які пов'язані між собою криптографічними методами [1]. Основний принцип технології полягає в тому, що кожен блок містить інформацію про попередній блок, створюючи безперервний ланцюг. Це забезпечує незмінність даних та високий рівень захисту від підробки або змін.

Криптографія відіграє ключову роль у функціонуванні блокчейну. Основою є геш-функції, що використовуються для хешування даних у блоці та створення цифрових підписів. Найпоширенішими алгоритмами є SHA-256 (Bitcoin) та Keccak-256 (Ethereum). Використання геш-функцій гарантує цілісність інформації, оскільки навіть мінімальні зміни у вхідних даних призводять до кардинальних змін у результаті хешування.

Таблиця 1.1 - Порівняльна характеристика алгоритмів консенсусу в блокчейн-системах

Алгоритм консенсусу	Принцип роботи	Переваги	Недоліки
Proof of Work (PoW)	Розв'язання криптографічних задач	Висока безпека	Велике енергоспоживання
Proof of Stake (PoS)	Вибір валідаторів на основі частки володіння	Менше енергоспоживання	Ризик централізації
Delegated Proof of Stake (DPoS)	Голосування за делегатів	Висока швидкість обробки	Необхідність довіри делегатам

Узгодженість транзакцій у блокчейні досягається через механізми консенсусу. Найпоширенішими алгоритмами є Proof of Work (PoW) та Proof of Stake (PoS)[4]. PoW використовується у Bitcoin і передбачає розв’язання складних обчислювальних задач для додавання нового блоку в ланцюг, що вимагає значних енергетичних витрат. PoS, який застосовується в Ethereum 2.0, базується на принципі підтвердження частки: право додавати блоки отримують

користувачі, які мають значну кількість токенів і ставлять їх як заставу. Це зменшує енергоспоживання та підвищує ефективність мережі.

Структура блокчейну складається з вузлів (ноди), які забезпечують децентралізоване збереження даних і виконання транзакцій. Вузли можуть бути повними (full nodes) та легкими (light nodes). Повні вузли зберігають увесь ланцюг блоків, тоді як легкі вузли зберігають лише заголовки блоків, що дозволяє їм економити ресурси.

Транзакції в блокчейні проходять через кілька етапів. Спочатку користувач створює транзакцію та підписує її закритим ключем. Потім вона передається в мережу та потрапляє в мемпул, де очікує підтвердження. Майнери або валідатори вибирають транзакції для включення в новий блок, перевіряють їхню коректність та додають до ланцюга після успішного виконання алгоритму консенсусу.

Смарт-контракти є важливою складовою сучасних блокчейн-систем. Це програмні алгоритми, що виконуються автоматично при виконанні певних умов. Вони дозволяють створювати децентралізовані додатки (DApps), автоматизувати бізнес-процеси та забезпечувати прозорість угод. Найпопулярнішою платформою для смарт-контрактів є Ethereum, що використовує мову програмування Solidity[6].

Блокчейн забезпечує децентралізацію, що означає відсутність єдиного центру керування. Це усуває необхідність довіряти третім сторонам і забезпечує підвищену стійкість до атак. Проте існують виклики, пов’язані з масштабованістю та швидкістю обробки транзакцій. У традиційних блокчейнах, таких як Bitcoin, обмежена кількість транзакцій на секунду, що призводить до затримок та високих комісій. Для розв’язання цієї проблеми розробляються рішення другого рівня, зокрема Lightning Network для Bitcoin та Rollups для Ethereum[7].

Безпека блокчейну ґрунтується на його криптографічних механізмах та механізмах консенсусу, але залишається вразливою до деяких атак. Найвідомішою є атака 51%, за якої зловмисник отримує контроль над більшістю

потужностей мережі та може змінювати історію транзакцій. Крім того, смарт-контракти можуть мати вразливості в коді, що використовується хакерами для викрадення коштів.

Розглянемо детальніше, як функціонує блокчейн на практиці. Для кращого розуміння цього процесу звернемося до Рисунку 1.1, який наочно демонструє принцип роботи блокчейну в криптовалютних транзакціях.

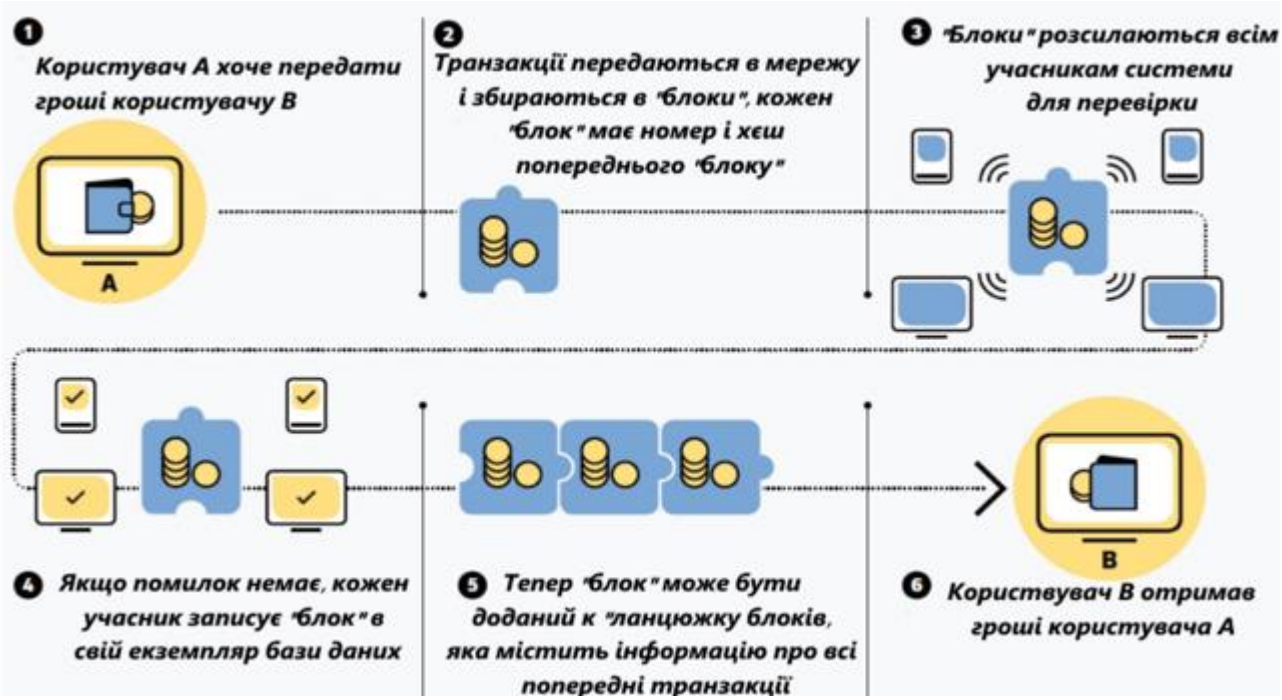

Рисунок 1.1 - Принцип роботи блокчейну в криптовалютних транзакціях

Як показано на рисунку 1.1 процес починається з бажання користувача А передати кошти користувачу В. Ця транзакція разом з іншими об'єднується в блоки, кожен з яких містить унікальний ідентифікатор і хеш попереднього блоку, що забезпечує цілісність всього ланцюжка. Потім ці блоки розсилаються всім учасникам мережі для верифікації. Якщо жодних помилок не виявлено, кожен вузол мережі записує новий блок у свою локальну копію бази даних, після чого блок додається до загального ланцюжка, що містить історію всіх попередніх транзакцій. В результаті цього процесу користувач В успішно отримує кошти від користувача А.

Однак, незважаючи на інноваційність технології блокчейн, вона має певні обмеження та виклики. Як уже зазначалося, популярні блокчейни, такі як Bitcoin, обмежені кількість транзакцій на секунду, що призводить до затримок та високих комісій. Для розв'язання цієї проблеми розробляються рішення другого рівня, зокрема Lightning Network для Bitcoin та Rollups для Ethereum.

Безпека залишається ключовим питанням для блокчейн-технологій. Хоча криптографічні механізми забезпечують високий рівень захисту, залишаються потенційні вразливості, такі як атака 51% або помилки в смарт-контрактах. Важливим аспектом є також енергоспоживання, особливо для блокчейнів, що використовують консенсус Proof of Work, який потребує значних обчислювальних потужностей. У відповідь на це розробляються більш екологічні алгоритми консенсусу, такі як Proof of Stake. Регуляторні виклики також становлять значний бар'єр для широкого впровадження блокчейну, оскільки різні країни мають різні підходи до регулювання цієї технології.

Незважаючи на виклики, блокчейн продовжує еволюціонувати, з'являються нові протоколи та платформи, які пропонують інноваційні рішення для існуючих проблем. Інтероперабельність між різними блокчейнами стає все важливішою для створення єдиної екосистеми децентралізованих додатків. Приватні та консорціумні блокчейни набувають популярності серед корпорацій, які прагнуть скористатися перевагами технології без повної децентралізації. Токенізація активів відкриває нові можливості для інвестицій та управління власністю в цифровому просторі.

DAO (децентралізовані автономні організації) пропонують новий підхід до корпоративного управління, що базується на прозорості та колективному прийнятті рішень[8]. З розвитком технологій квантових обчислень виникають нові виклики для криптографічних алгоритмів, що лежать в основі блокчейну, стимулюючи дослідження квантово-стійких рішень.

Блокчейн-технологія продовжує розвиватися та вдосконалюватися, впроваджуючи нові механізми та підходи. Розглянемо додаткові аспекти архітектури та принципів роботи.

Рисунок 1.2 - Базова структура блокчейну: послідовне з'єднання блоків через хеш-посилання

На рисунку 1.2 представлена базова структура блокчейну, яка демонструє послідовний зв'язок блоків у ланцюжку. Кожен блок містить хеш попереднього блоку, дані транзакцій та часову мітку (timestamp). Така структура забезпечує неможливість змінити історію транзакцій, оскільки зміна даних в одному блоці призведе до зміни його хешу, що порушить цілісність усього ланцюжка. Блоки розташовані в хронологічному порядку від найновішого (Блок N) до попередніх (Блок N-1, Блок N-2), формуючи безперервний ланцюг, де кожен наступний блок підтверджує достовірність попереднього.

Однак із зростанням популярності блокчейн-мереж виникають проблеми масштабування. Шардинг є одним з ключових рішень для масштабування блокчейну. Це процес розділення мережі на менші частини (шарди), кожна з яких обробляє певний набір транзакцій паралельно. Така архітектура дозволяє значно збільшити пропускну здатність мережі, оскільки різні групи вузлів можуть одночасно обробляти різні транзакції.

На Рисунок 1.3 зображено архітектуру шардингу, де основний ланцюг (Beacon Chain) координує роботу декількох шардів (Шард 1, Шард 2, Шард 3). Кожен шард містить власні транзакції, стан та валідатори, але при цьому залишається пов'язаним з основним ланцюгом, що забезпечує загальну узгодженість системи. Таке рішення дозволяє розподілити навантаження між різними частинами мережі, зберігаючи при цьому її безпеку та децентралізацію.

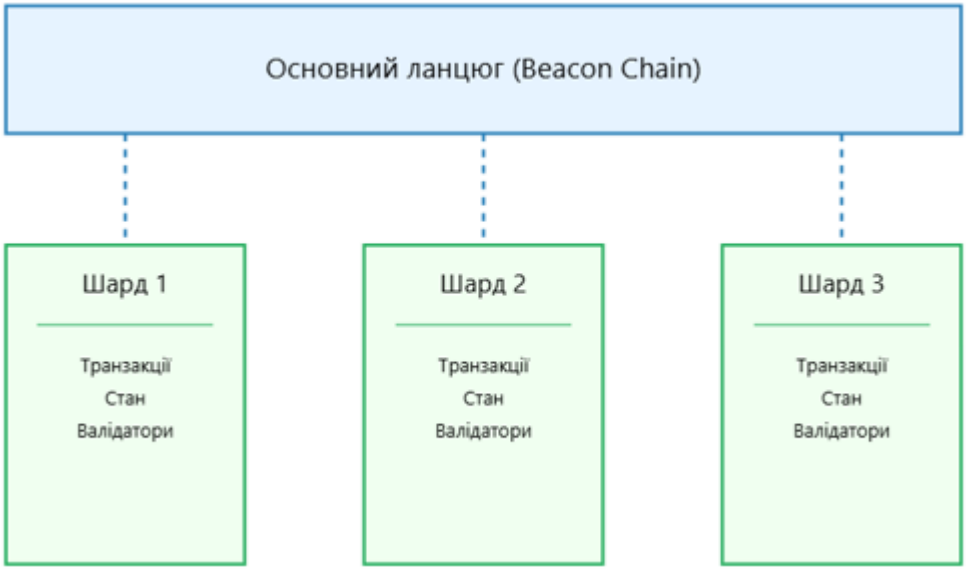

Рисунок 1.3 - Архітектура шардингу в блокчейні: взаємодія основного ланцюга (Beacon Chain) з шардами

Впровадження шардингу є важливим кроком у розвитку блокчейн-технологій, особливо для таких платформ як Ethereum 2.0. Це дозволяє вирішити трилему блокчейну, забезпечуючи одночасно безпеку, децентралізацію та масштабованість. Завдяки таким інноваціям блокчейн стає все більш придатним для широкомасштабного впровадження в різних галузях, від фінансів до управління ланцюгами постачання, відкриваючи нові можливості для цифрової трансформації суспільства.

Механізми забезпечення конфіденційності в блокчейні реалізуються через різні криптографічні протоколи. Одним з найважливіших є Zero-Knowledge Proof (ZKP) - метод, який дозволяє одній стороні довести іншій, що твердження є істинним, не розкриваючи жодної додаткової інформації [7].

Окрім структурних особливостей блокчейну, важливим аспектом цієї технології є забезпечення конфіденційності та приватності транзакцій, що стає все більш критичним у сучасному цифровому світі.

Рисунок 1.4 - Криптографічні методи забезпечення приватності в блокчейн-транзакціях

На Рисунок 1.4 представлені різні криптографічні методи, що використовуються в блокчейн-системах для забезпечення приватності. Верхня схема демонструє процес Zero-Knowledge Proof, який дозволяє підтверджувати достовірність інформації без розкриття самої інформації, середня схема показує механізм Ring Signature для анонімізації транзакцій через змішування підписів, а нижня схема ілюструє процес конфіденційних транзакцій від приховування суми до публікації зашифрованих даних у блокчейн.

Розвиток криптографічних методів та архітектурних рішень, таких як шардинг і Layer 2, відіграє ключову роль у подоланні обмежень традиційних блокчейн-систем, забезпечуючи необхідний баланс між децентралізацією, безпекою, масштабованістю та приватністю. Впровадження цих інноваційних технологій відкриває шлях до широкого використання блокчейну в різноманітних сферах, від фінансів до державного управління, створюючи основу для нової децентралізованої цифрової економіки та перетворюючи способи обміну цінностями та даними у глобальному масштабі.

1.2 Аналіз вразливостей та загроз безпеки блокчейн-систем

Блокчейн-технологія, незважаючи на свою інноваційність та високий рівень безпеки, має ряд вразливостей і загроз, які можуть бути використані зловмисниками для порушення роботи системи, крадіжки коштів або маніпуляції даними. Однією з основних загроз є атака 51%. Ця атака можлива, коли одна особа або група осіб контролює більше 50% обчислювальної потужності мережі, що дозволяє їм маніпулювати блокчейном, наприклад, виключати або змінювати порядок транзакцій. Така атака особливо актуальна для блокчейнів з невеликою кількістю учасників або низьким рівнем децентралізації.

Іншою серйозною загрозою є атаки на смарт-контракти. Смарт-контракти — це програмний код, який виконується в блокчейні, і якщо в цьому коді є помилки або вразливості, зловмисники можуть їх використати для крадіжки коштів або порушення логіки роботи контракту. Прикладом таких атак є інцидент з DAO (Decentralized Autonomous Organization) в Ethereum, де через вразливість у смарт-контракті було вкрадено кошти на суму понад 50 мільйонів доларів[8]. Фішингові атаки також є поширеною загрозою для користувачів блокчейн-систем. Зловмисники можуть створювати підроблені веб-сайти або розсилати електронні листи, щоб обманом отримати доступ до приватних ключів користувачів. Оскільки приватні ключі є основним способом доступу до криптовалютних гаманців, їх втрата може призвести до повної втрати коштів.

Ще однією вразливістю є проблеми з масштабованістю та затримками в мережі. Збільшення кількості транзакцій може призвести до перевантаження мережі, що робить її більш уразливою до атак, таких як DDoS (розподілені атаки типу «відмова в обслуговуванні»). Такі атаки можуть тимчасово паралізувати роботу блокчейну, ускладнюючи проведення транзакцій або взагалі роблячи мережу недоступною. Помилки в реалізації консенсусних алгоритмів також можуть стати причиною серйозних проблем. Наприклад, у разі використання алгоритму Proof of Stake (PoS) можлива атака «нічого на кону» (Nothing at Stake), коли валідатори можуть підтримувати кілька різних версій блокчейну

одночасно, що призводить до розгалуження ланцюга та втрати консенсусу. Крім того, існує загроза атак на рівні мережевого протоколу. Наприклад, атака Sybil, коли зловмисник створює велику кількість підроблених ідентифікаторів у мережі, щоб отримати контроль над нею. Це може призвести до маніпуляцій з транзакціями або навіть до повного захоплення мережі.

Проблеми з приватним ключем також є серйозною загрозою. Якщо користувач втрачає доступ до свого приватного ключа, він втрачає доступ до своїх коштів назавжди, оскільки відновити ключ неможливо. Це підкреслює важливість надійного зберігання та резервного копіювання приватних ключів. Крім того, існують загрози, пов'язані з квантовими обчисленнями. У майбутньому квантові комп'ютери можуть стати достатньо потужними, щоб зламати криптографічні алгоритми, які використовуються в блокчейні, такі як ECDSA (Elliptic Curve Digital Signature Algorithm). Це може призвести до того, що зловмисники зможуть підроблювати підписи та отримувати доступ до коштів користувачів.

Соціальна інженерія також є серйозною загрозою. Зловмисники можуть використовувати психологічні методи, щоб обманом отримати доступ до конфіденційної інформації, такої як приватні ключі або паролі. Наприклад, вони можуть видавати себе за представників служби підтримки або інших довірених осіб. Проблеми з регулюванням та правовою базою також можуть вплинути на безпеку блокчейн-систем. Відсутність чітких регуляторних рамок може призвести до зловживань з боку учасників мережі, а також ускладнити боротьбу з кіберзлочинністю. Нарешті, важливо враховувати загрози, пов'язані з інтеграцією блокчейну з іншими системами. Наприклад, якщо блокчейн використовується в поєднанні з централізованими системами, вразливості в цих системах можуть стати точкою входу для атак на весь блокчейн.

Як було показано в розділі 1.1, структура блокчейну забезпечує високий рівень безпеки через криптографічні механізми та децентралізацію. Однак, навіть така архітектура не гарантує повної захищеності від усіх видів атак. Особливо це стосується публічних блокчейнів, які мають найвищий рівень

децентралізації, але водночас найбільш уразливі до атак, спрямованих на користувачів і смарт-контракти.

Для подолання проблем масштабованості, як було описано раніше, розробляються такі рішення як шардинг (Рисунку 1.3 - Архітектура шардингу в блокчейні: взаємодія основного ланцюга (Beacon Chain) з шардами) та Layer 2 рішення. Ці підходи, хоч і вирішують проблему масштабованості, створюють нові виклики з точки зору безпеки, зокрема, потенційні атаки на шарди або на механізми взаємодії між різними рівнями блокчейну.

Захист приватності транзакцій також є важливим аспектом безпеки блокчейн-систем. На Рисунку 1.4 показано різні криптографічні методи, що використовуються в блокчейн-системах для забезпечення приватності, включаючи Zero-Knowledge Proof, Ring Signature та конфіденційні транзакції. Ці методи допомагають захистити дані користувачів, але також можуть створювати додаткову складність і потенційні вразливості в системі.

Таким чином, незважаючи на високий рівень безпеки, блокчейн-системи мають ряд вразливостей, які потребують уваги та постійного вдосконалення для забезпечення надійності та стабільності їх роботи. Забезпечення безпеки блокчейн-систем вимагає комплексного підходу, що охоплює як технічні аспекти (удосконалення протоколів, підвищення стійкості смарт-контрактів, впровадження надійних криптографічних методів), так і організаційні заходи (навчання користувачів, розробка стандартів безпеки, створення механізмів швидкого реагування на інциденти).

Серед важливих аспектів безпеки блокчейн-систем особливу увагу слід приділити смарт-контрактам, які виступають одним із найуразливіших компонентів, оскільки помилки в їхній розробці можуть призвести до значних фінансових втрат, як у випадку інциденту з DAO.

На Рисунок 1.5 представлено життєвий цикл смарт-контракту в блокчейн-системі, що охоплює чотири ключові етапи: створення, розгортання, виконання та завершення. Етап створення включає переговори між зацікавленими сторонами та розробку коду смарт-контракту з подальшою його валідацією.

Розгортання передбачає збереження контракту в блокчейні та блокування цифрових активів учасників. На етапі виконання відбувається автоматична перевірка умов договору та запуск відповідних операторів контракту. Завершальний етап характеризується оновленням стану учасників, виділенням та розморожуванням цифрових активів.

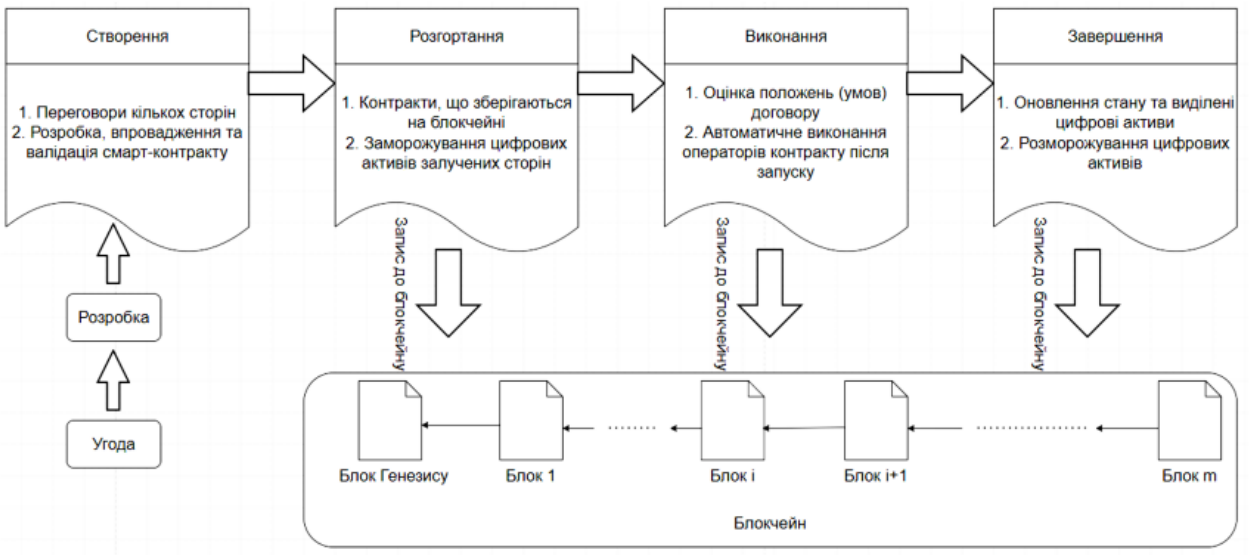

Рисунок 1.5 - Життєвий цикл смарт-контракту в блокчейн-системі

Кожен із цих етапів записується в блокчейні як окрема транзакція, починаючи від блоку генезису та продовжуючись через усі наступні блоки. Така структура забезпечує прозорість та незмінність усіх дій, пов'язаних зі смарт-контрактом, але водночас підкреслює важливість ретельної розробки та тестування коду, оскільки після розгортання внести зміни практично неможливо.

1.3 Сучасні методи захисту криптовалютних транзакцій

Сучасні методи захисту криптовалютних транзакцій представляють комплексний підхід до забезпечення безпеки в блокчейн-системах, що має безпосередній зв'язок із загальною кібербезпекою. Як зазначено у тексті, криптовалютні транзакції захищаються різноманітними технологіями, серед яких асиметричне шифрування, цифрові підписи (ECDSA, EdDSA), методи

підвищення анонімності (CoinJoin, Zero-Knowledge Proofs), мультипідписи та апаратні гаманці.

Особливу увагу приділено таким криптографічним методам як кільцеві підписи (Ring Signature), приховані адреси (Stealth Addresses) та протоколи конфіденційності (MimbleWimble), що суттєво ускладнюють аналіз транзакцій у блокчейні. Ці технології співзвучні з методами, представленими на Рисунку 1.4 і забезпечують надійний захист особистих даних користувачів.

Важливе місце у документі займають Layer-2 рішення (Lightning Network, zk-Rollups), захист від мережевих атак та механізми розподіленого управління ключами (Shamir's Secret Sharing, Social Recovery Wallets). Ці підходи відображають еволюцію методів безпеки в блокчейн-системах та їх адаптацію до нових загроз.

Дослідження у сфері розслідування кіберінцидентів охоплюють широкий спектр підходів, методологій та технологій. Серед значущих праць варто відзначити роботу Cichonski та ін. (2012) «Комп'ютерна безпека: посібник з обробки інцидентів» для NIST, яка заклала фундаментальні принципи реагування на інциденти кібербезпеки[9]. Johnson (2021) у своїй праці «Cybersecurity Incident Response: How to Contain, Eradicate, and Recover from Incidents» запропонував структурований підхід до розслідування та реагування на інциденти, підкреслюючи важливість документування та навчання[10].

У контексті блокчейн-технологій та криптовалютних систем варто згадати дослідження Conti та ін. (2022) «A Survey of Security Threats in Blockchain Systems»[11], де автори аналізують специфічні кіберзагрози для блокчейн-систем та методи їх розслідування. Zhang та Wang (2019) у праці «Security and Privacy in Blockchain-Based Systems» [12] висвітлюють проблеми безпеки та захисту приватності в блокчейн-системах, пропонуючи методи виявлення та реагування на інциденти безпеки.

Впровадження штучного інтелекту та машинного навчання у процеси кібербезпеки досліджено у роботі Kaloudi та Li (2020) «The AI-Based Cyber Threat Landscape», [13] де автори аналізують перспективи та виклики використання ШІ

для виявлення та протидії кіберзагрозам. Moustafa та ін. (2022) у дослідженні «Intelligent Machine Learning-Based Cybersecurity Incident Response Framework» [14] запропонували інтеграцію ML/AI для автоматизації процесів реагування на інциденти.

Важливий внесок у розуміння методологій розслідування інцидентів зробили Ceccato і Newton (2025) у праці «System thinking for sustainable crime prevention», [15] де представлено системний підхід до аналізу та запобігання кіберзлочинам. У роботі Захаріана та Петрової (2024) «Алгоритми автоматизованого розслідування кіберінцидентів з використанням штучного інтелекту» запропоновано інноваційні підходи до автоматизації процесів розслідування з використанням ШІ.

Попередні дослідження зосереджувались на психологічних профілях зловмисників та механізмах виникнення інцидентів, однак сучасні кіберзагрози вимагають розробки адаптивних методів реагування на їх постійну еволюцію. Серед найнебезпечніших атак 2022-2025 років виділяються таргетований фішинг з використанням ШІ, багаторівневі DDoS-атаки, комплексні SQL-ін'єкції, програми-вимагачі з подвійним шифруванням, атаки на ланцюги постачання та експлуатація вразливостей нульового дня [3]. Ці загрози потребують спеціалізованих підходів до розслідування та реагування, що враховують особливості кожного типу атак та дозволяють ефективно протидіяти сучасним кіберзагрозам.

Ефективна кіберзахисна стратегія вимагає не лише якісних методів передачі запитів до служби IT-безпеки, але й комплексних систем управління інцидентами. Організації можуть використовувати різноманітні захищені канали комунікації, від телефонних ліній з обов'язковою верифікацією до спеціалізованих порталів самообслуговування з багатофакторною автентифікацією. Ці канали доповнюються централізованими системами управління, такими як Security Operation Center (SOC) та Incident Response Platform (IRP), що забезпечують цілодобовий моніторинг, оперативний аналіз загроз та автоматизацію процесів розслідування інцидентів)[3].

Після впровадження відповідних систем реалізується чітко структурований процес обробки запитів безпеки, що охоплює чотири послідовні етапи: реєстрацію, аналіз, реагування та закриття. На етапі реєстрації черговий спеціаліст або автоматизована система створює заявку та визначає її пріоритет. Потім аналітики безпеки проводять оцінку загрози та визначають напрямок розслідування, після чого команда реагування здійснює нейтралізацію загрози з використанням спеціалізованих інструментів. Завершальний етап передбачає аналіз першопричин інциденту та оновлення бази знань, що забезпечує постійне вдосконалення системи кіберзахисту організації.

У рамках модернізованого підходу до кіберзахисту розроблено удосконалений алгоритм розслідування інцидентів (Рисунок 1.6).

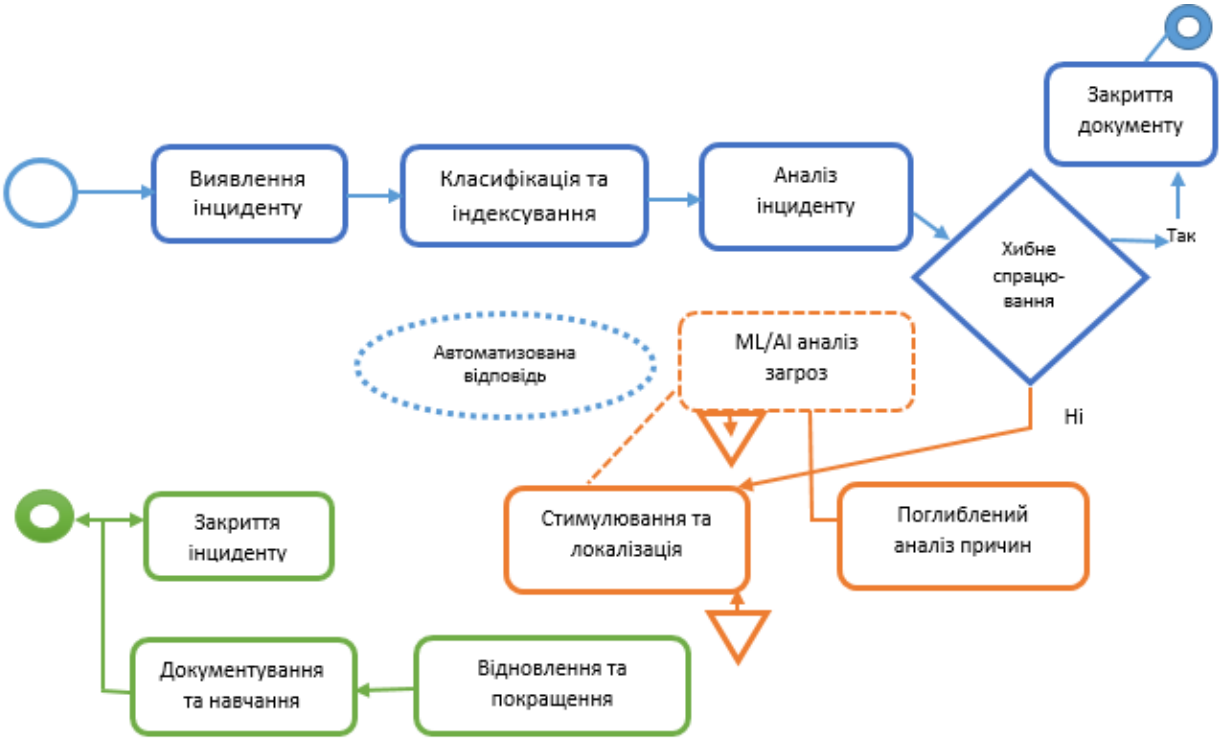

Рисунок 1.6 - Алгоритм розслідування інцидентів

План реагування на інциденти забезпечує структурований підхід до кіберзагроз через етапи виявлення, аналізу та відновлення, з чітким розподілом ролей та відповідальності команди, що мінімізує простої та фінансові втрати. Удосконалена схема на 2025-2030 роки впроваджує ML/AI аналіз загроз та автоматизовану відповідь, створюючи інтегрований підхід для сучасного

захисту організацій, з перспективою розробки адаптивних механізмів реагування на новітні кіберзагрози.

Такий підхід дозволяє ефективно адаптуватися до таких сучасних загроз як таргетований фішинг з використанням ШІ, багаторівневі DDoS-атаки, комплексні SQL-ін'єкції, програми-вимагачі з подвійним шифруванням, атаки на ланцюги постачання та експлуатація вразливостей нульового дня. Інтеграція технологій машинного навчання та штучного інтелекту в процеси розслідування інцидентів створює адаптивну систему безпеки, здатну еволюціонувати паралельно з розвитком загроз. Впровадження даного алгоритму дозволить організаціям мінімізувати час реагування на інциденти, зменшити фінансові втрати від кібератак та підвищити загальний рівень захищеності критичної інфраструктури в умовах сучасних кіберконфліктів.

Висновки до розділу 1

У першому розділі проведено комплексне дослідження теоретичних основ безпеки блокчейн-технологій, що дозволило сформувати цілісне розуміння архітектури, принципів функціонування та методів захисту сучасних блокчейн-систем. Аналіз показав, що безпека забезпечується децентралізованою архітектурою з криптографічними методами (хеш-функції, цифрові підписи), механізмами консенсусу (PoW, PoS) та комплексом захисних заходів, включаючи Zero-Knowledge Proofs та Layer 2 рішення. Виявлено основні вразливості блокчейн-систем та обґрунтовано необхідність інтеграції алгоритмів машинного навчання для ефективного виявлення сучасних кіберзагроз та мінімізації часу реагування на інциденти.

2 КРИПТОГРАФІЧНІ МЕТОДИ ТА АЛГОРИТМИ ЗАХИСТУ БЛОКЧЕЙН-ТРАНЗАКЦІЙ

2.1 Математичні основи криптографічного захисту транзакцій

Криптографічний захист блокчейн-транзакцій ґрунтується на фундаментальних математичних концепціях, які забезпечують конфіденційність, цілісність та автентичність даних. Хеш-функції є базовим криптографічним примітивом у блокчейн-системах. Криптографічна хеш-функція $H: \{0,1\} * \rightarrow \{0,1\}^{n}$ відображає вхідні дані довільної довжини у вихідні дані фіксованої довжини n біт. Для блокчейн-систем критично важливими є наступні властивості криптографічної хеш-функції:

1) стійкість до знаходження прообразу (preimage resistance):

$$P(x: H(x) = h) < 2^{-n} \text{ для будь-якого } h; \quad (2.1)$$

2) стійкість до колізій (collision resistance):

$$P(x_1, x_2: H(x_1) = H(x_2) \wedge x_1 \neq x_2) < 2^{-n}/^2 \quad (2.2)$$

Вибір конкретної хеш-функції визначає рівень стійкості системи до атак методом повного перебору та колізій. У таблиці 2.1 наведено порівняльний аналіз основних хеш-функцій, що використовуються у провідних блокчейн-системах.

Таблиця 2.1 - Порівняння криптографічних хеш-функцій

Алгоритм	Розмір хешу (біт)	Блокчейн	Складність атаки
SHA-256	256	Bitcoin	2^{128}
Keccak-256	256	Ethereum	2^{128}
BLAKE2b	512	Zcash	2^{256}
SHA-3	256	Various	2^{128}

Цифрові підписи на еліптичних кривих (ECDSA) забезпечують автентичність транзакцій. Еліптична крива над полем Fp визначається рівнянням еліптичної кривої:

$$E: y^2 = x^3 + ax + b\ (mod\ p) \qquad (2.4)$$

де $a, b \in Fp$ та $4a^3 + 27b^2 \neq 0\ (mod\ p)$

Алгоритм ECDSA складається з трьох етапів (рисунок 2.1): генерації ключової пари (приватний ключ $d \in [1, n-1]$, публічний ключ $Q = d \times G$, створення підпису через обчислення хешу $e = H(M)$, випадкового числа k та компонентів $r = (k \times G)_x\ mod\ n$, s = k^{-1}(e + dr) *mod* n, і верифікації отриманої пари (r, s).

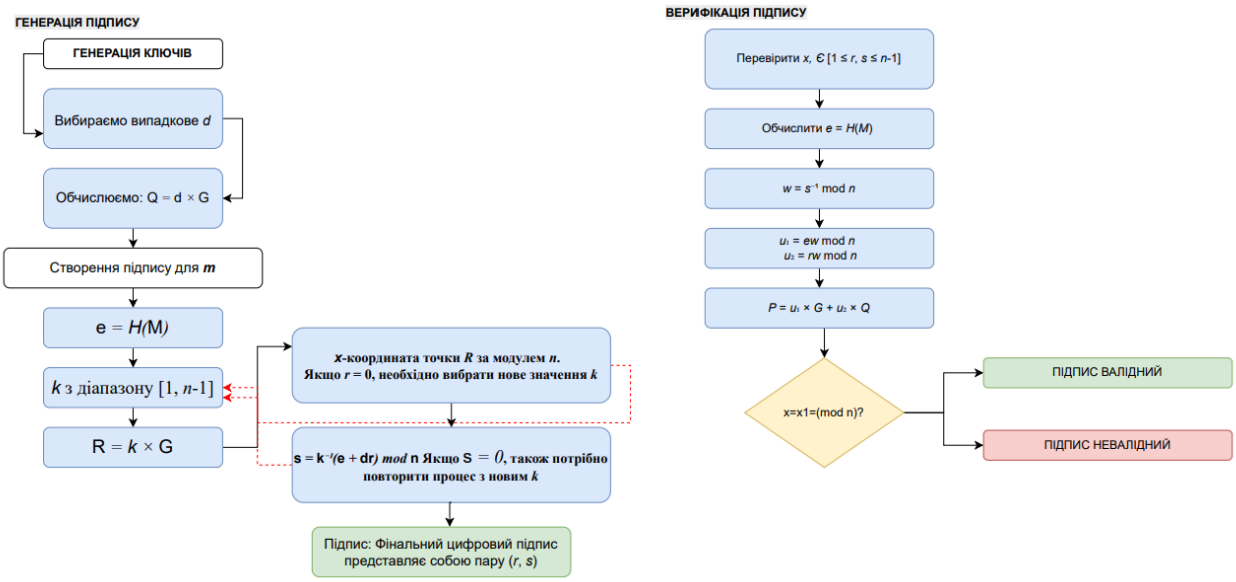

Рисунок 2.1 - Генерація та верифікація цифрового підпису ECDSA

Безпека алгоритму ECDSA базується на обчислювальній складності розв'язання задачі дискретного логарифмування на еліптичних кривих (ECDLP), для якої не існує субекспоненційних алгоритмів розв'язання. Це дозволяє використовувати значно коротші ключі при збереженні еквівалентного рівня безпеки порівняно з класичними методами у скінченних полях. Для кривої з n-бітним порядком складність атак становить:

- дискретний логарифм (ECDLP): $O(\sqrt{n}) \approx O(2^{n}/^{2})$ операцій;

- відновлення приватного ключа: $O(2^{n})$ операцій;

- підбір підпису: $O(2^{n})$ операцій.

Блокчейн-системи використовують стандартизовані криві з перевіреними параметрами безпеки, які пройшли ретельний криптографічний аналіз. Найпоширенішою є крива secp256k1, розроблена консорціумом SEC та прийнята Bitcoin у 2009 році, а згодом Ethereum та іншими платформами з параметрами:

$$p = 2^{256} - 2^{32} - 2^{9} - 2^{8} - 2^{7} - 2^{6} - 2^{4} - 1 \qquad (2.5)$$

де a = 0, b = 7, що дає рівняння $y^2 = x^3 + 7 \ (mod\ p)$. Ця крива забезпечує 128-бітний рівень безпеки при використанні 256-бітних ключів, що достатньо для захисту від усіх відомих атак, але вразливе до майбутніх квантових комп'ютерів. Коефіцієнти рівняння еліптичної кривої $y^2 = x^3 + 7 \ (mod\ p)$. Спрощена форма з $a = 0$ прискорює обчислення.

G = (79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798,

483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8)

Базова точка (генератор) n - фіксована точка на кривій, яка використовується для генерації всіх публічних ключів. Координати x та y записані у шістнадцятковому форматі. Множення приватного ключа d на точку G дає публічний ключ $Q = d \times G$.

n = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141

Порядок базової точки G - кількість різних точок, які можна отримати послідовним додаванням G до себе. Це просте число визначає максимальну

кількість можливих приватних ключів ($\sim 2^{256}$) і є критичним параметром для безпеки системи.

Ці параметри є публічними константами, спільними для всіх користувачів Bitcoin та Ethereum, що забезпечує сумісність та можливість верифікації підписів будь-ким у мережі.

Таблиця 2.2 - Порівняння еліптичних кривих для цифрових підписів

Параметр	secp256k1	Ed25519	secp256r1
Розмір ключа	256 біт	256 біт	256 біт
Рівень безпеки	128 біт	128 біт	128 біт
Швидкість підпису	Середня	Висока	Середня
Швидкість верифікації	Середня	Дуже висока	Середня
Стійкість до сайд-каналів	Низька	Висока	Середня
Використання	Bitcoin, Ethereum	Algorand, Stellar	TLS, IPSec

ECDSA має обмеження щодо агрегації множинних підписів у компактну форму, що збільшує розмір та вартість multisig транзакцій.

Схема Шнорра представляє альтернативу з лінійною властивістю агрегації, де для публічних ключів $P_1, P_2, \ldots, P_\text{n}$ та відповідних підписів $\sigma_1, \sigma_2, \ldots, \sigma_\text{n}$:

$$\sigma_{agg} = \sigma_1 + \sigma_2 + \ldots + \sigma_\text{n} \tag{2.6}$$

$$P_{agg} = P_1 + P_2 + \ldots + P_\text{n} \tag{2.7}$$

де σ_{agg} є валідним підписом для P_{agg}.

Пряма агрегація вразлива до «rogue key attack», тому розроблено протокол MuSig, який використовує криптографічне хешування для генерації унікальних коефіцієнтів. MuSig забезпечує $n - \text{з} - n$ багатопідписну схему, де всі учасники беруть участь у підписанні, зберігаючи компактність схеми Шнорра. Це дозволяє створювати агрегований підпис розміром як один індивідуальний підпис, але з підтвердженням згоди всіх учасників.

Протокол MuSig складається з трьох фаз: агрегації публічних ключів через обчислення $L = H(Q_1, Q_2, \ldots, Q_n)$ та коефіцієнтів $a_i = H(L, Q_i)$, для формування: $\tilde{Q} = a^1 Q^1 + a_2 Q_2 + \ldots + a_n Q_n$.; створення підпису через генерацію (nonce) k_i, обчислення $R_i = k_i \times G$, агрегації $R = R_1 + R_2 + \ldots + R_n$. та часткових підписів $s_i = k_i + c \cdot a_i \cdot d_i$; агрегації у фінальний підпис (R, s) де $s = s_1 + s_2 + \ldots + s_n$. Верифікація виконується перевіркою рівності $s \times G = R + c \times \tilde{Q}$, де потрібен лише агрегований ключ без знання окремих учасників. MuSig має константну складність $O(1)$~64 байти незалежно від кількості учасників, на відміну від лінійного зростання $O(n)$ в ECDSA multisig, забезпечуючи приватність та масштабованість для Lightning Network та корпоративних блокчейнів. Математична структура кожного раунду представлена як композиція послідовних перетворень $state_{i+1} = MC\left(SR\left(S(state_i)\right)\right) \oplus$ над станом блоку, де кожна операція послідовно застосовується для досягнення максимальної криптографічної стійкості.

Симетричне шифрування AES працює з блоками 128 біт через 10-14 раундів залежно від розміру ключа, кожен раунд виконує операції SubBytes, ShiftRows, MixColumns та AddRoundKey для забезпечення криптографічної стійкості.

Таблиця 2.3 - Режими роботи блокового шифрування AES

Режим	Паралелізація	Випадковий доступ	Автентичність
CBC	Тільки дешифрування	Ні	Ні
CTR	Так	Так	Ні
GCM	Так	Так	Так
XTS	Так	Так	Ні

Для шифрування повідомлень довільної довжини використовують режими роботи блокового шифру (таблиця 2.3), що забезпечують різні компроміси між продуктивністю, безпекою та функціональністю, включаючи можливість паралелізації, випадкового доступу та автентичності. Режим GCM поєднує

шифрування з автентичністю, CTR дозволяє паралелізацію та випадковий доступ, тоді як CBC забезпечує базове шифрування з обмеженими можливостями.

У криптографії діє принцип «найслабшої ланки»: загальна безпека системи визначається її найуразливішим елементом, оскільки атакуючий зосередить зусилля саме на ньому. Загальна стійкість S системи з n компонентів обчислюється як:

$$S = min(S_1, S_2, \ldots, S_n) \quad (2.8)$$

де S_i- стійкість i-го компонента в бітах;

Для Bitcoin загальна стійкість становить $S(\text{система}) = min(S(SHA256),$ S(ECDSA-secp256k1)) = min(128, 128) = 128 біт, що достатньо для захисту від усіх практичних атак. Збільшення стійкості лише одного компонента не підвищить загальну безпеку системи, якщо інші залишаться на рівні 128 біт, оскільки атакуючий завжди обиратиме найслабший елемент. З появою квантових комп'ютерів ця стійкість може бути скомпрометована, що вимагає переходу на квантово-стійкі алгоритми у майбутньому.

2.2 Алгоритми консенсусу та їх безпека

Механізми консенсусу вирішують проблему Byzantine Generals Problem у розподіленому середовищі без довіреної третьої сторони, забезпечуючи узгодженість стану блокчейн-мережі при наявності зловмисних або несправних вузлів. Різні механізми пропонують власні компроміси між безпекою, продуктивністю, енергоефективністю та децентралізацією для функціонування без централізованого координатора. Proof of Work (PoW) базується на обчислювальній складності знаходження nonce, що задовольняє умову валідності:

$$H(block\_header \,||\, nonce) < target \quad (2.9)$$

де $target = 2^{256} / difficulty$.

Майнери змагаються у пошуку значення, яке при хешуванні дає результат менший за встановлений поріг, створюючи асиметрію між складністю створення блоку та тривіальністю верифікації. Ця асиметрія робить атаки економічно невигідними, оскільки атакуючий повинен контролювати більшість обчислювальної потужності мережі, що вимагає надмірних інвестицій порівняно з потенційними вигодами.

Параметр *difficulty* динамічно коригується мережею для підтримки сталого часу генерації блоків незалежно від зміни обчислювальної потужності - чим менше значення *target*, тим більше провідних нулів потрібно у хеші. Процес пошуку *nonce* є ітеративним та імовірнісним без аналітичного методу обчислення, тому майнери послідовно перебирають мільярди варіантів до знаходження валідного розв'язку. Майнер з часткою α обчислювальної потужності знаходить блок з ймовірністю α, що забезпечує справедливий розподіл винагороди пропорційно до внеску в безпеку мережі. З імовірнісної точки зору знаходження валідного блоку є випадковим процесом, де кожна спроба хешування є незалежним експериментом Бернуллі з ймовірністю успіху $p = target / 2^{256}$. Кількість спроб підкоряється геометричному розподілу з математичним сподіванням $E[n] = 1/p$, що дозволяє прогнозувати середній час майнінгу блоку. Для Bitcoin ймовірність успіху становить приблизно 10^{-22}, що робить процес надзвичайно ресурсоємним. Криптографічні хеш-функції генерують псевдовипадкові значення з рівномірним розподілом, тому майнінг моделюється як послідовність незалежних випробувань. Це дозволяє передбачити очікуваний час знаходження блоку залежно від обчислювальної потужності майнера та поточної складності мережі (рисунок 2.2).

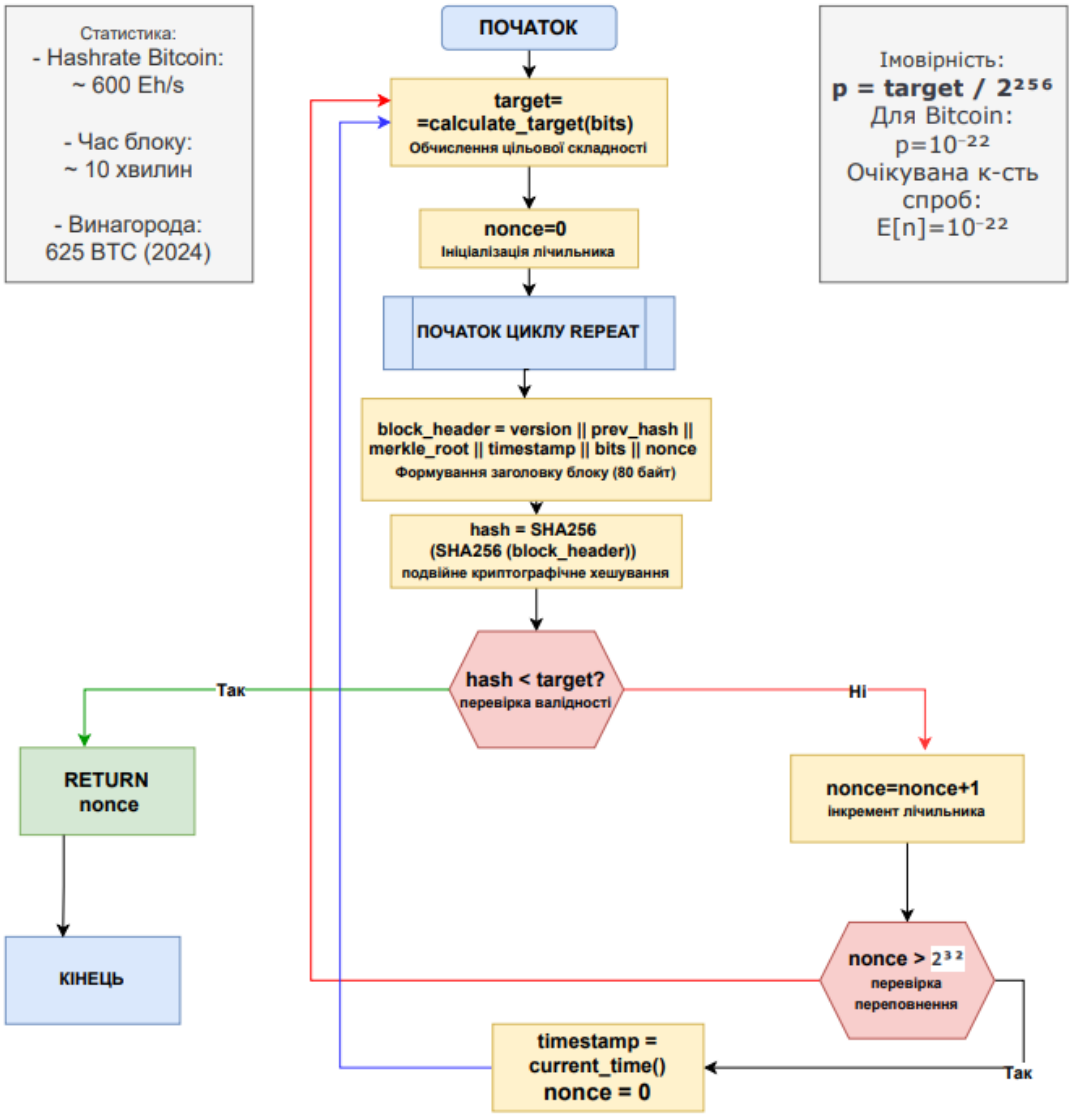

Рисунок 2.2 - Алгоритм Proof of Work

Ймовірність знаходження блоку:

$$P(\text{успіх}) = 1 - \left(\frac{1}{\text{difficulty}}\right)^{\text{hash\_rate} \times \text{time}} \quad (2.10)$$

очікуваний час:

$$T = \text{difficulty} / \text{hash\_rate} \quad (2.11)$$

difficulty – складність мережі;

hash_rate – швидкість хештегування;

time – проміжок часу;

P(успіх) – ймовірність знайти блок за час time;

T – математичне сподівання часу знаходження блоку.

Коригування складності Bitcoin відбувається кожні 2016 блоків за формулою:

$$\text{difficulty\_new} = \text{difficulty\_old} \times (2016 \times 10\ \text{min}) / \text{actual\_time} \quad (2.12)$$

де 2016 - кількість блоків між коригуваннями

Практичні параметри різних блокчейн-платформ демонструють значні відмінності: Bitcoin використовує SHA-256 з блоками кожні 10 хвилин, Ethereum (до переходу на PoS) працював з Ethash та блоками 12-14 секунд, а Litecoin застосовує Scrypt з блоками кожні 2,5 хвилини (таблиця 2.4).

Таблиця 2.4 - Параметри Proof of Work у різних блокчейнах

Параметр	Bitcoin	Ethereum (до PoS)	Litecoin
Час блоку	10 хв	12-14 с	2.5 хв
Алгоритм хешування	SHA-256	Ethash	Scrypt
Розмір блоку	~1-2 MB	~15 KB	~1-2 MB
Коригування difficulty	2016 блоків	Кожен блок	2016 блоків
Винагорода (2024)	3.125 BTC	N/A	6.25 LTC

Механізм автоматичного коригування забезпечує стабільність часу генерації блоків, але не гарантує захист від атаки 51%, коли атакуючий контролює більшість хешрейту.

PoW забезпечує високу безпеку та децентралізацію, але його енергоспоживання ~150 TWh/рік та пропускна здатність 7-15 TPS стали критичними обмеженнями. Proof of Stake (PoS) замінює обчислювальну роботу економічним стейкінгом з вибором валідаторів за формулою

$$P(V_i) = \frac{\text{stake}_i}{\sum_i \text{stake}_j} \quad (2.13)$$

де:

- stake_i- частка валідатора i (кількість застейканих токенів);
- $\sum_i \text{stake}_j$ - загальна застейкана сума всіх валідаторів у мережі.

Ethereum 2.0 Casper FFG організовує валідацію в епохи по 6,4 хвилини з мінімальним стейком 32 ETH та системою штрафування за зловмисну поведінку. Delegated Proof of Stake (DPoS) дозволяє токенхолдерам голосувати за делегатів (21-101), досягаючи >1000 TPS за рахунок часткової централізації. Порівняльний аналіз показує кардинальні відмінності: PoW має найвище енергоспоживання (~150 TWh/рік) та найнижчу пропускну здатність (7-15 TPS), PoS забезпечує баланс безпеки та ефективності (0.01 TWh/рік, 1000-10000 TPS), а DPoS досягає найвищої швидкості за рахунок зниженої децентралізації (таблиця 2.5).

Таблиця 2.5 - Порівняльний аналіз механізмів консенсусу

Характеристика	PoW	PoS	DPoS
Енергоспоживання	Дуже високе (~150 TWh/рік)	Низьке (~0.01 TWh/рік)	Низьке
Децентралізація	Висока	Середня-висока	Низька-середня
Пропускна здатність	7-15 TPS	1000-10000 TPS	1000-4000 TPS
Фінальність	Імовірнісна	Детермінована	Детермінована
Вхідний бар'єр	Високий (ASIC)	Середній (32 ETH)	Низький (голосування)
Атака 51%	Контроль хешрейту	Контроль стейку + slashing	Контроль делегатів
Приклади	Bitcoin, Litecoin	Ethereum 2.0, Cardano	EOS, Tron

Byzantine Fault Tolerance (BFT) протоколи ефективні для приватних блокчейнів, де Practical BFT (PBFT) забезпечує консенсус при $f < n/3$ Byzantine вузлів. PBFT працює через три фази (Pre-prepare, Prepare, Commit) з комунікаційною складністю $O(n^2)$ та латентністю 2 раунди. Механізм View Change забезпечує відмовостійкість при збої лідера через трансляцію VIEW-CHANGE повідомлень.

Безпека різних механізмів формалізується для PoW ймовірність атаки:

$$P(attack\_success) = (\alpha / (1 - \alpha))^k \quad (2.14)$$

Для PoS при $f < n/3$ атака практично неможлива через механізми slashing та детермінованої фінальності:

$$P(attack\_success) \approx 0, \text{якщо } f < n/3 \quad (2.15)$$

PoW створює імовірнісну фінальність, тоді як PoS та BFT забезпечують детерміновану фінальність з економічними штрафами за зловмисну поведінку. Кожен механізм має власні компроміси між безпекою, продуктивністю, енергоефективністю та децентралізацією залежно від конкретного застосування.

2.3 Протоколи Zero-Knowledge Proofs для приватності транзакцій

Традиційні блокчейн-системи забезпечують повну прозорість транзакцій, що створює проблеми конфіденційності для користувачів та організацій при збереженні потреби у верифікації коректності операцій. Zero-Knowledge Proofs (ZKP) дозволяють доказувачу переконати верифікатора в істинності твердження без розкриття додаткової інформації - можна довести валідність транзакції (достатність коштів, коректність підписів) без розкриття сум чи адрес. Інтерактивний протокол $\langle P, V \rangle$ є zero-knowledge, якщо існує симулятор S, що генерує розподіл $S(x)$, обчислюється формулою (2.16)

$$VIEW\_V(P(x), V(\text{x}))^{**} \quad (2.16)$$

означаючи відсутність витоку інформації понад факт істинності. Надійний ZKP повинен задовольняти три властивості: повноту (чесний доказувач майже завжди переконує верифікатора), коректність (зловмисний доказувач не може довести

хибне твердження) та власне zero-knowledge властивість (відсутність витоку секретної інформації).

Надійний ZKP повинен задовольняти три властивості:

- повноту $P(V\ accepts \mid P\ honest\ \wedge\ x\ \in\ L)\ \geq\ 1 - \varepsilon$;
- коректність $P(V\ accepts \mid P\ dishonest\ \wedge\ x\ \notin\ L)\ \leq\ \delta^*$;
- zero-knowledge $\forall V, \exists S$: $VIEW\_V(\langle \mathrm{P(x)}, \mathrm{V^*(x)} \rangle)\ \approx$ _c $\mathrm{S(x)}^{**}$.

де ε, δ - експоненційно малі функції від параметра безпеки. Ці властивості гарантують, що чесний доказувач майже завжди переконає верифікатора, зловмисний доказувач не може довести хибне твердження, а симулятор може згенерувати нерозрізненний транскрипт без секретної інформації. Протокол Шнорра є класичним прикладом інтерактивного ZKP для доказу знання дискретного логарифму - фундаментальної задачі криптографічних систем на еліптичних кривих.

Протокол Шнорра для доказу знання дискретного логарифму працює з публічними параметрами: просте число p, підгрупа порядку q (тобто $q \mid (p-1)$), генератор g підгрупи, де кожен користувач має секретний ключ $x \in \mathbb{Z}_q$ і відповідний публічний ключ $y = g^x\ mod\ p$, з метою довести знання x без його розкриття верифікатору. Протокол складається з чотирьох етапів: на етапі Commit доказувач P вибирає випадкове $r \leftarrow \mathbb{Z}_q$, обчислює commitment $R = g^r mod\ p$ та надсилає R верифікатору V; на етапі Challenge верифікатор V вибирає випадкове challenge $c \leftarrow \mathbb{Z}_q$ та надсилає c доказувачу P; на етапі Response доказувач P обчислює response $s = r + cx\ mod\ q$ та надсилає s верифікатору V.

На етапі Verification верифікатор V перевіряє рівність $g^s = R \cdot y^c\ mod\ p$, приймаючи доказ якщо рівність виконується та відхиляючи в іншому випадку, що підтверджує знання доказувачем секретного ключа x без його безпосереднього розкриття. Доведення zero-knowledge властивості демонструє, що симулятор S може генерувати розподіл нерозрізненний від справжнього протоколу, вибираючи випадкові $s \leftarrow \mathbb{Z}_q$ та $c \leftarrow \mathbb{Z}_q$, і обчислюючи $R = g^s \cdot y^{(-c)}$, де цей підроблений транскрипт (R, c, s) є статистично

нерозрізненим від справжнього, що доводить відсутність витоку інформації про секретний ключ x під час виконання протоколу.

Протокол Шнорра демонструє елегантність інтерактивних ZKP, однак вимагає обміну повідомленнями, що ускладнює застосування у блокчейн-системах з асинхронною верифікацією тисячами вузлів. zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments) вирішують ці проблеми через неінтерактивність (один доказ) та компактність з константним часом верифікації незалежно від складності обчислення. Основа zk-SNARKs - Quadratic Arithmetic Programs (QAP), що перетворюють довільне обчислення у поліноміальні обмеження через формулу:

$$\exists \text{ поліноми } a(x), b(x), c(x): a(x) \cdot b(x) - c(x) = h(x) \cdot Z(x) \qquad (2.24)$$

де $Z(x) = (x - r_1)(x - r_2)\ldots(x - r_1\, r_n)$ – цільовий поліном.

QAP дозволяє звести перевірку складного обчислення до верифікації однієї поліноміальної тотожності через паринги на еліптичних кривих без розкриття значень поліномів, але потребує складної церемонії довіреної установки (trusted setup).

Перетворення обчислення у поліноміальну форму є лише першим кроком. Для створення криптографічно безпечного доказу необхідна складна церемонія довіреної установки (trusted setup) та використання парингів на еліптичних кривих для приховування значень поліномів при збереженні можливості верифікації їх властивостей.

Алгоритм zk-SNARK складається з трьох фаз: Setup генерує Common Reference String (CRS) з секретними параметрами τ,α,β,γ,δ («toxic waste»), які після використання повинні бути знищені для запобігання фальшивим доказам; Prove обчислює компоненти доказу $\pi = (A, B, C)$ розміром ~200 байт через комбінації поліномів у секретній точці τ; Verify перевіряє парингову рівність $e(A, B) = e(\alpha G_1, \beta G_2) \cdot e(IC, \gamma G_2) \cdot e(C, \delta G_2)$ за ~10 мс без розкриття значень поліномів. Основний ризик - trusted setup з «toxic waste», тому використовуються

багатосторонні церемонії (MPC), де достатньо одного чесного учасника для безпеки системи. Groth16 має найкомпактніші докази з окремим setup для кожного обчислення, тоді як PLONK дозволяє універсальний setup для різних схем, підвищуючи гнучкість впровадження. zk-STARKs усувають trusted setup через хеш-функції та забезпечують квантову стійкість, але мають розмір доказу 100-300 KB замість 200 байт, що робить їх придатними для zkRollups.

Bulletproofs займають проміжну позицію з розміром доказу 1-2 KB без trusted setup, оптимальні для range proofs - доказів значень у діапазоні без розкриття самого значення. Вибір протоколу залежить від пріоритетів: мінімальний розмір та швидка верифікація - zk-SNARKs, прозорість setup та квантова стійкість - zk-STARKs, компактні range proofs без trusted setup - Bulletproofs (таблиця 2.6).

Таблиця 2.6 - Порівняння протоколів Zero-Knowledge Proofs

Характеристика	zk-SNARKs	zk-STARKs	Bulletproofs
Розмір доказу	~200 bytes	~100-300 KB	~1-2 KB
Час верифікації	~10 ms	~20-50 ms	~100-300 ms
Trusted setup	Потрібен	Не потрібен	Не потрібен
Квантова стійкість	Ні	Так	Ні
Використання	Zcash, Filecoin	StarkWare	Monero, Mimblewimble

Практичні застосування демонструють різні підходи: Zcash використовує zk-SNARKs для повної конфіденційності транзакцій, StarkWare розробляє zkRollups з економією простору блокчейну у 100-1000 разів, а Monero застосовує Bulletproofs зі зменшенням розміру транзакцій на 80%. Розвиваються universal schemes (PLONK, Halo 2) та гібридні підходи, що комбінують переваги різних ZKP, а спеціалізовані апаратні прискорювачі зменшують час генерації доказів для масового впровадження технологій приватності у блокчейн-системах.

Висновки до розділу 2

У другому розділі проведено комплексний аналіз криптографічних методів захисту блокчейн-транзакцій, що охоплює математичні основи, механізми консенсусу та протоколи Zero-Knowledge Proofs. Дослідження показало, що фундаментальною основою безпеки є криптографічні хеш-функції SHA-256 для Bitcoin та Keccak-256 для Ethereum, які забезпечують 128-бітний рівень захисту через властивості стійкості до колізій та знаходження прообразу. Цифрові підписи ECDSA на еліптичній кривій secp256k1 гарантують автентичність транзакцій, однак схема Шнорра з протоколом MuSig вирішує проблему агрегації множинних підписів, забезпечуючи константний розмір незалежно від кількості учасників.

Аналіз механізмів консенсусу виявив кардинальні відмінності: PoW забезпечує максимальну безпеку при енергоспоживанні 150 TWh/рік та пропускній здатності 7-15 TPS, тоді як PoS досягає балансу з енергоспоживанням 0,01 TWh/рік та продуктивністю 1000-10000 TPS. Протоколи Zero-Knowledge Proofs представляють революційний підхід до приватності: zk-SNARKs забезпечують найкращу ефективність з доказами 200 байт за 10 мс верифікації, але вимагають довіреної установки, тоді як zk-STARKs усувають цю вимогу та гарантують квантову стійкість при більшому розмірі доказів. Практичні застосування підтверджують успішне впровадження цих технологій у Zcash для повної конфіденційності, Monero для приховування сум та StarkWare для масштабування через zkRollups.

Загальна стійкість системи визначається принципом найслабшої ланки, що вимагає збалансованого підходу до вибору криптографічних компонентів та підготовки до квантових загроз у майбутньому.

3. СИСТЕМА МОНІТОРИНГУ ТА ВИЯВЛЕННЯ АНОМАЛІЙ У КРИПТОВАЛЮТНИХ МЕРЕЖАХ

3.1 Алгоритми машинного навчання для детекції кіберзагроз

Криптографічні методи та алгоритми консенсусу, розглянуті у розділі 2, забезпечують фундаментальну безпеку блокчейн-транзакцій на протокольному рівні, однак навіть найдосконаліші криптографічні примітиви не можуть запобігти всім типам кіберзагроз, особливо тим, що експлуатують людський фактор, вразливості смарт-контрактів або складні багатокрокові схеми атак. Фішингові атаки, Ponzi-схеми, відмивання коштів та маніпуляції ринком часто є технічно валідними з точки зору блокчейн-протоколу, але зловмисними з точки зору безпеки користувачів та регуляторних вимог, що вимагає додаткового рівня захисту через інтелектуальну систему моніторингу, здатну аналізувати поведінкові патерни, виявляти аномалії та класифікувати загрози у реальному часі.

Виявлення кіберзагроз у блокчейн-мережах вимагає використання передових алгоритмів машинного навчання, здатних обробляти великі обсяги транзакційних даних у реальному часі та адаптуватися до еволюції атак, оскільки традиційні системи на основі правил виявляються неефективними проти нових типів атак та складних схем, що постійно модифікуються зловмисниками. Застосування моделей машинного навчання дозволяє автоматизувати виявлення аномалій, класифікацію загроз та прогнозування потенційних інцидентів безпеки через навчання на історичних даних та адаптацію до нових патернів зловмисної поведінки.

Система виявлення базується на багатошаровій архітектурі, що включає препроцесинг даних, екстракцію ознак, навчання моделей та постпроцесинг результатів, де основою підходу є обчислення аномальності для кожної транзакції або адреси, що відображає ступінь відхилення від нормальної поведінки згідно з моделлю виявлення аномалій (3.1).

$$A(x) = \begin{cases} 1, \text{якщо } S(x) > \theta \text{ (аномалія)} \\ 0, \text{інакше (норма)} \end{cases} \quad (3.1)$$

де:

- $S(x)$ - оцінка аномальності для об'єкта x;
- θ - порогове значення для класифікації.

Структуру взаємодії компонентів цієї моделі наочно представлено на рисунку 3.1, який ілюструє процес обробки транзакції від етапу введення вхідних даних до прийняття остаточного рішення про класифікацію.

Рисунок 3.1 - ER-діаграма моделі виявлення аномалій у блокчейн-транзакціях

Ефективність моделей машинного навчання критично залежить від якості вхідних даних та екстракції релевантних ознак, оскільки сирі блокчейн-транзакції містять лише базову інформацію (адреси відправника/отримувача, суму, часову мітку), яка сама по собі недостатня для виявлення складних атак та вимагає збагачення додатковим контекстом через статистику поведінки адрес,

темпоральні патерни, графові метрики та взаємозв'язки з відомими злочинними адресами.

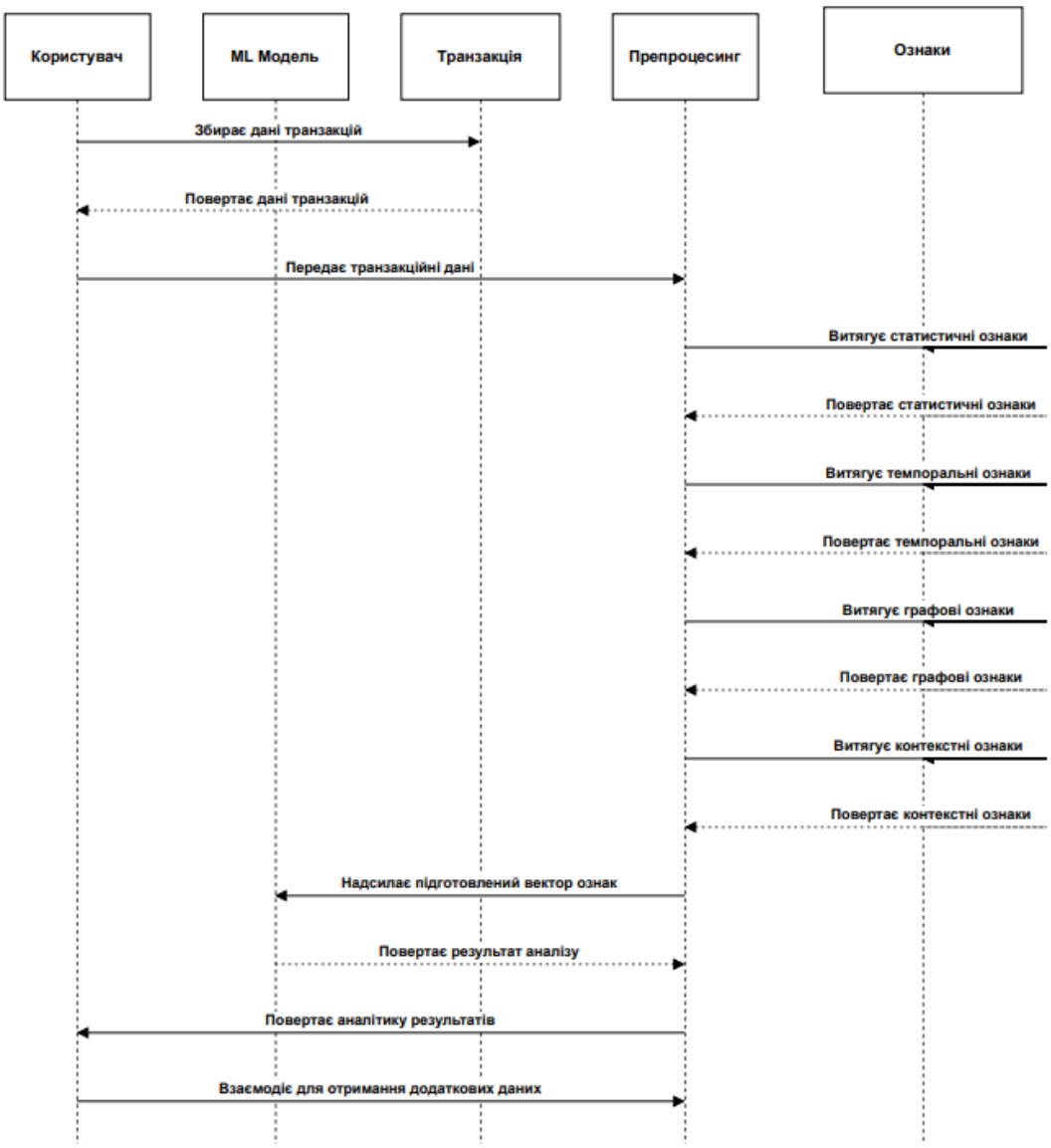

Рисунок 3.2 – Діаграма послідовності препроцесингу блокчейн-транзакцій

Діаграма послідовності на рисунку 3.2 ілюструє процес препроцесингу блокчейн-транзакцій для машинного навчання через взаємодію п'яти компонентів. Процес починається зі збору даних транзакцій від Користувача до компонента Транзакція, після чого транзакційні дані передаються до модуля Препроцесинг для послідовної екстракції чотирьох типів ознак через вертикальні взаємодії з компонентом Ознаки.

Модуль Препроцесинг виконує чотири послідовні операції витягування та повернення ознак: статистичні ознаки (ступені входу/виходу, баланс, середня

сума транзакцій), темпоральні ознаки (частота транзакцій, інтервали між ними, час життя адреси), графові ознаки (PageRank, коефіцієнт кластеризації, центральність проміжності) та контекстні ознаки (взаємодія з біржами, зв'язки з фішинговими адресами), де кожна операція показана суцільною стрілкою для запиту та пунктирною для повернення результату. Після завершення екстракції всіх ознак виконується нормалізація вектора за формулою (3.2)

$$X_{norm} = (X - \mu) / \sigma \quad (3.2)$$

де μ - середнє, σ - стандартне відхилення

Z-score нормалізація забезпечує, що всі ознаки мають однаковий масштаб, що критично для алгоритмів на основі відстаней.

Підготовлений нормалізований вектор ознак Препроцесинг надсилає до ML Модель для аналізу, яка повертає результат аналізу назад через ланцюжок компонентів до Користувача, а пунктирні лінії на діаграмі представляють повернення результатів або зворотний зв'язок між компонентами, тоді як суцільні лінії показують прямі запити або передачу даних у багатоетапному конвеєрі препроцесингу. Результуючий вектор містить понад 20 ознак, що комплексно характеризують поведінку адреси та забезпечують надійну основу для подальшої класифікації загроз за допомогою моделей машинного навчання з можливістю виявлення фішингових атак, Ponzi-схем та відмивання коштів через аналіз багатовимірних патернів поведінки[35].

Isolation Forest [36] є ефективним алгоритмом для виявлення аномалій у багатовимірних даних, що базується на принципі легшої ізоляції аномалій від решти даних через менше випадкових розділень простору ознак, (рисунок 3.3) складається з двох основних кроків: побудови дерева ізоляції (iTree) та обчислення оцінки аномальності (anomaly score). Крок побудови дерева рекурсивно створює бінарне дерево через випадковий вибір ознаки q з простору *features* та випадкового значення розділення p між мінімальним та

максимальним значеннями цієї ознаки, розділяючи дані на X_{left} для точок менших за p та X_{right} для точок більших або рівних p, де процес зупиняється при досягненні обмеження висоти або коли залишається одна точка, що створює зовнішній вузол.

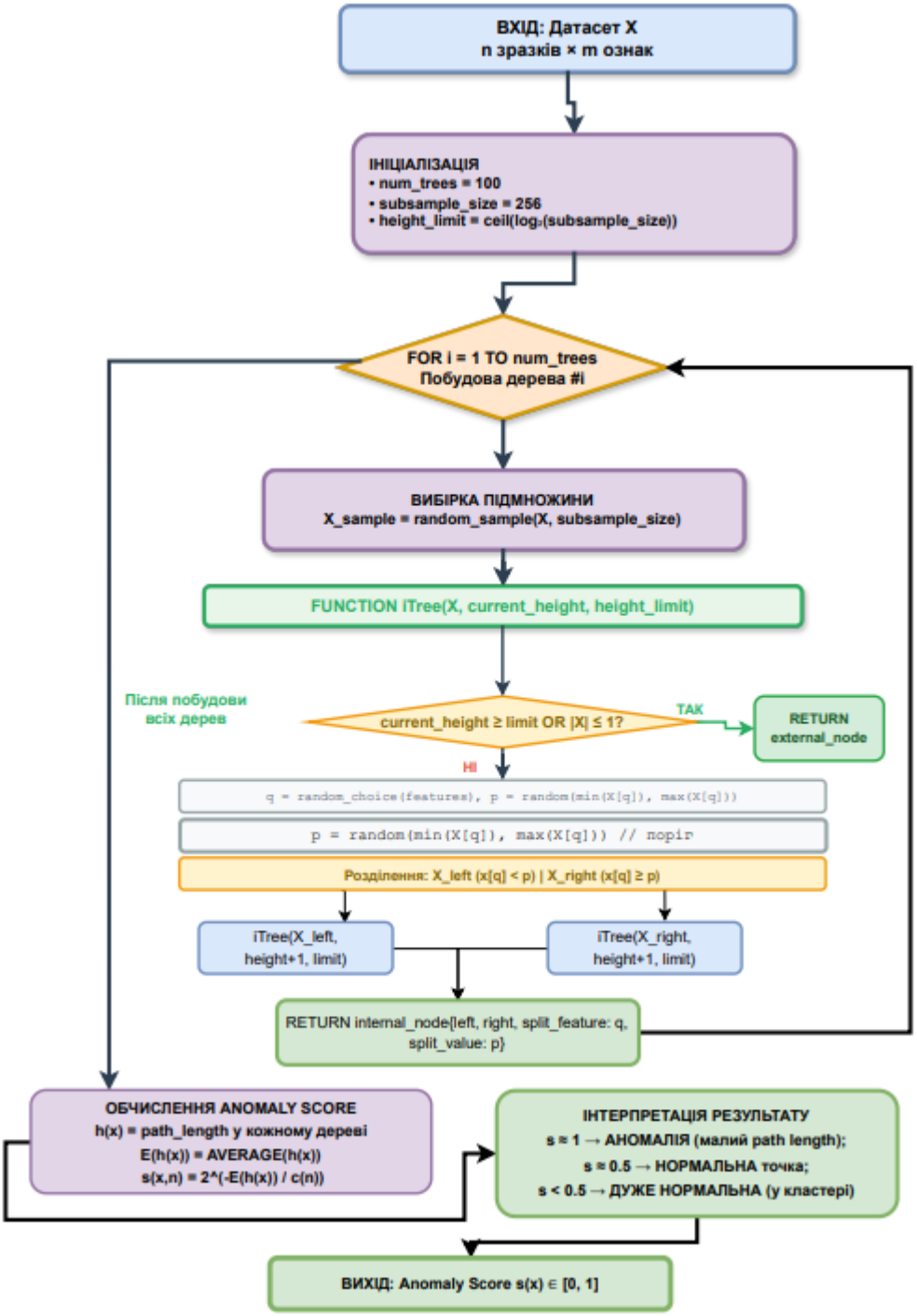

Рисунок 3.3 - Блок-схема алгоритму Isolation Forest для виявлення аномалій

Крок обчислення оцінки аномальності визначає довжину шляху $h(x)$ для точки x у кожному дереві, обчислює середнє значення $E(h(x))$ по всіх деревах, та розраховує фінальну оцінку за формулою (3.3)

$$s(x,n) = 2^{(-E(h(x)) / c(n))} \quad (3.3)$$

де:

$c(n) = 2H(n-1) - 2(n-1)/n,$ з константою Ейлера $H(i) \approx ln(i) + 0.5772,$ що дозволяє нормалізувати оцінку відносно розміру датасету. Інтерпретація результатів показує, що значення s близько до 1 вказує на аномалію з малою довжиною шляху, s близько до 0.5 означає нормальну точку, а s менше 0.5 представляє дуже нормальну точку глибоко у кластері, що забезпечує інтуїтивну шкалу для прийняття рішень про класифікацію.

Ключова перевага Isolation Forest полягає у його обчислювальній ефективності — алгоритм має лінійну складність $O(n)$ відносно кількості точок, що робить його придатним для аналізу великих блокчейн-датасетів у реальному часі. На відміну від методів на основі відстаней (*k-NN, DBSCAN*), які мають квадратичну складність $O(n^2)$, Isolation Forest масштабується ефективно навіть для мільйонів транзакцій. Процес роботи алгоритму візуалізовано на рисунку 3.3, що ілюструє етапи побудови ізоляційних дерев через випадкові розділення простору ознак, обчислення довжин шляхів для тестової точки у кожному дереві ансамблю, та формування фінального anomaly score на основі усередненої глибини ізоляції для виявлення аномальних блокчейн-транзакцій.

Застосування Isolation Forest до блокчейн-транзакцій дозволяє виявляти складні багатовимірні патерни аномальної поведінки, які важко формалізувати через прості правила. Алгоритм ефективно ідентифікує як окремі підозрілі транзакції, так і координовані атаки, що включають множину пов'язаних адрес.

Isolation Forest ефективний для unsupervised виявлення аномалій, але не може класифікувати конкретні типи загроз, тому для розрізнення фішингу, Ponzi-схем та відмивання коштів потрібен supervised learning підхід через

Gradient Boosting [36], що будує потужний класифікатор через послідовне додавання слабких learners. Gradient Boosting є потужним supervised learning алгоритмом, що будує ансамбль слабких класифікаторів через послідовне додавання моделей, де кожна наступна модель фокусується на корекції помилок попередніх, приймаючи на вхід тренувальний набір $D = \{(x_i, y_i)\}_{i=1}^{n}$ та функцію втрат $L\big(y, F(x)\big)$, і повертає фінальну модель $F_M(x)$.

Алгоритм починається з ініціалізації базової моделі, що мінімізує функцію втрат:

$$F_0(x) = \arg min_\gamma \sum_{i=1}^{n} L(y_i, \gamma) \tag{3.4}$$

що мінімізує функцію втрат, після чого виконує M ітерацій, де на кожній ітерації обчислюються градієнти функції втрат відносно поточних передбачень за формулою (3.5)

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F=F_{m-1}} \tag{3.5}$$

та вказують напрямок найбільшого зменшення помилки для кожного зразка. Тренується базовий learner (зазвичай Decision Tree обмеженої глибини) $h_\mathrm{m}(x)$ на даних $(x_i,\ r_{im})$, що апроксимує псевдо-залишки та навчається передбачати корекції для помилок поточної моделі, після чого визначається оптимальний коефіцієнт $\gamma_m = \arg min_\gamma \sum_{i=1}^{n} L\big(y_i, F_{m-1}(x_i) + \gamma\ h_m(x_i)\big)$ через line search для забезпечення оптимального внеску нового дерева у фінальну модель.

Модель оновлюється додаванням зваженого внеску нового дерева $F_m(x) = F_{m-1}(x) + v \cdot \gamma_m \cdot h_m(x)$, де $v\ \in\ (0,1]$— коефіцієнт навчання (learning rate), що контролює швидкість навчання, де менші значення $v\ (0.01 - 0.1)$ забезпечують кращу генералізацію, але вимагають більше ітерацій. Після M ітерацій повертається фінальна модель $F_m(x)$, що представляє адитивний ансамбль базових learners, де для отримання ймовірностей класів застосовується сигмоїдальна функція $p = \sigma\big(F_m(x)\big) = \frac{1}{1+e^{-F_m(x)}}$.

ї

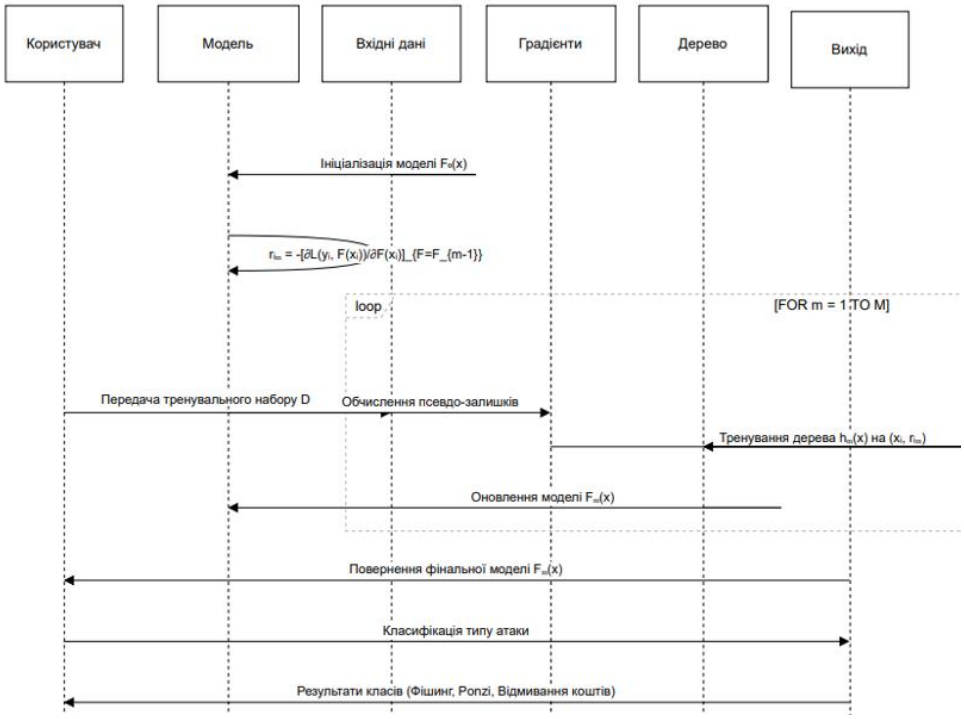

Рисунок 3.4 - Діаграма послідовності виконання алгоритму Gradient Boosting

Ключова перевага цього підходу полягає в послідовній корекції помилок, де кожне нове дерево спеціалізується на тих зразках, де попередні моделі помилялися, що призводить до високої точності класифікації та дозволяє ефективно розрізняти складні типи атак у блокчейн-безпеці, навіть якщо їхні ознаки частково перетинаються. Детальну послідовність виконання алгоритму Gradient Boosting із взаємодією основних компонентів системи представлено на рисунку 3.4, що ілюструє повний цикл від ініціалізації моделі через ітеративне навчання до фінальної класифікації типу атаки, дозволяючи не лише виявити аномальну транзакцію, а й визначити конкретний тип атаки з високою точністю для автоматизованого реагування та формування адекватних контрзаходів.

Теоретичні алгоритми потребують емпіричної валідації на реальних даних для оцінки їх практичної ефективності, тому проведено порівняльне тестування п'яти ML-моделей на датасеті з 500 000 Ethereum адрес, включаючи 50 000 фішингових з Etherscan Phishing Database, де результати (таблиця 3.1) демонструють класичний компроміс між точністю та швидкістю навчання: Neural Network досягає найвищої точності 95.2% та F1-score 0.943 за 25.4 хвилини, XGBoost забезпечує оптимальний баланс 94.7% точності за 8.7 хвилини

для production систем, Isolation Forest показує найвищий recall 94.3% за 2.8 хвилини для unsupervised сценаріїв, а LightGBM демонструє найшвидше навчання 3.1 хвилини при точності 93.8% для систем з обмеженими ресурсами. Розглянуті алгоритми (Isolation Forest, Gradient Boosting) аналізують статичні ознаки адрес, ігноруючи темпоральну структуру послідовностей транзакцій, однак багато типів зловмисної поведінки мають характерні часові патерни: Ponzi-схеми демонструють експоненційне зростання активності з наступним колапсом, ботнети створюють регулярні періодичні транзакції, а wash trading показує циклічні патерни купівлі-продажу.

Таблиця 3.1 - Порівняння ML-моделей для детекції фішингових адрес

Модель	Точність	Recall	F1-score	Час навчання
Random Forest	92.3%	89.7%	0.910	5.2 хв
XGBoost	94.7%	92.1%	0.934	8.7 хв
LightGBM	93.8%	91.4%	0.926	3.1 хв
Neural Network	95.2%	93.5%	0.943	25.4 хв
Isolation Forest	88.1%	94.3%	0.911	2.8 хв

Long Short-Term Memory (LSTM) [37] мережі ефективні для аналізу послідовностей транзакцій у часі, виявляючи складні темпоральні залежності через механізм воріт (gates), що контролює потік інформації через клітини пам'яті та дозволяє виявляти багатоетапні атаки, такі як поступове виведення коштів при exit scam або послідовні транзакції у схемах відмивання, які неможливо детектувати через аналіз окремих транзакцій. LSTM використовує систему воріт для контрольованого збереження та забування інформації у довгостроковій пам'яті, де кожна комірка обробляє вхідний вектор x_t та попередній прихований стан h_t, оновлюючи стан пам'яті C_t через ворота забування f_t, ворота входу i_t, значення-кандидати $\tilde{C}_t$, оновлення стану комірки C_t та ворота виходу O_t для визначення прихованого стану h_t.

Ворота забування $f_t = \sigma\left(W_f \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b_f\right)$ приймає рішення, яку інформацію з попереднього стану C_{t-1} слід забути. Значення близькі до 0 означають «забути», близькі до 1 — «зберегти». Наприклад, при зміні патерну поведінки адреси (перехід від нормальних транзакцій до підозрілих), f_t може обнулити нерелевантну інформацію з попереднього контексту.

Ворота входу $i_t = \sigma\left(W_i \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b_i\right)$ контролює, які нові значення слід записати у стан пам'яті. Він працює у парі значення-кандидатами $\tilde{C}_t$, що містять нові потенційні значення для збереження.

Значення кандидати $\tilde{C}_t = tanh(W_c) \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b_c$ — це нові значення-кандидати, згенеровані на основі поточного входу та попереднього прихованого стану через tanh активацію (діапазон [-1,1]).

Оновлення стану комірки $C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$ — центральна «конвеєрна стрічка» пам'яті, що оновлюється комбінацією забутої старої інформації ($f_t \odot C_{t-1}$) та нових записаних значень ($i_t \odot \tilde{C}_t$). Це дозволяє LSTM зберігати інформацію на довгих часових інтервалах без градієнтного затухання.

Ворота виходу $O_t = \sigma(W_0) \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b_0$ визначає, яку частину оновленого стану комірки слід вивести як прихований стан h_t, який передається наступному timestep та використовується для класифікації.

Архітектуру та процес роботи LSTM для аналізу послідовностей блокчейн-транзакцій детально відображено на рисунку 3.5.

Процес навчання LSTM на блокчейн-даних [38].

1) Підготовка послідовностей: для адреси з транзакціями $[tx_1, tx_2, \ldots, tx_n]$ створюються ковзні вікна «*sliding windows*» фіксованого розміру w:

$$X = \begin{bmatrix} \begin{bmatrix} tx_{1,} & \cdots, & tx_w \\ \vdots & \ddots & \vdots \\ tx_{n-w+1} & \cdots & tx_n \end{bmatrix} \end{bmatrix} \tag{3.6}$$

Наприклад, для $w = 10$ кожна послідовність містить 10 послідовних транзакцій. Це дозволяє моделі «бачити» еволюцію поведінки адреси у часі.

2) forward pass через LSTM layers: кожна послідовність обробляється через один або кілька LSTM шарів. На кожному $timestep\ t$ LSTM оновлює свій внутрішній стан згідно з формул, акумулюючи інформацію про всю послідовність. Багатошаровий LSTM (stacked LSTM) дозволяє вивчати ієрархічні темпоральні представлення.

3) Dense layer для класифікації: фінальний прихований стан h_n (або послідовність станів для sequence-to-sequence задач) подається на повнозв'язний (Dense) шар з softmax активацією для класифікації типу загрози.

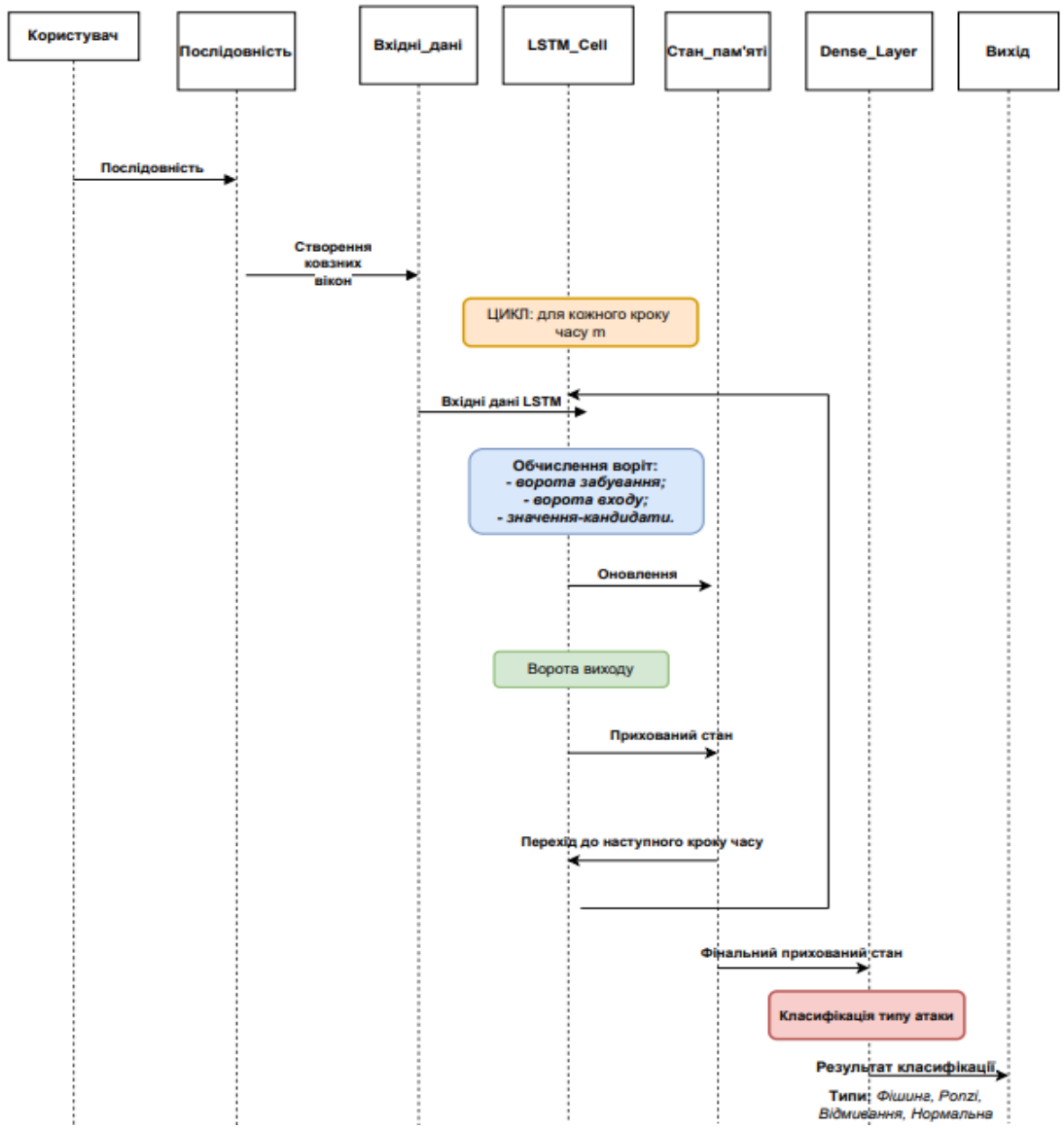

Рисунок 3.5 - Архітектура LSTM для аналізу послідовностей транзакцій

4) Backpropagation Through Time (BPTT): градієнти обчислюються через «розгортання» LSTM у часі, поширюючись назад через всю послідовність. LSTM архітектура з воротами мінімізує проблему зникаючих градієнтів, що дозволяє ефективно навчатися на довгих послідовностях (100+ timesteps).

5) Оптимізація (Adam optimizer): використовується adaptive learning rate optimizer Adam з типовими параметрами (learning_$rate = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$), що автоматично адаптує швидкість навчання для кожного параметра моделі.

Застосування LSTM до блокчейн-транзакцій дозволяє виявляти складні темпоральні патерни, такі як:

- поступове нарощування частоти транзакцій перед exit scam;
- циклічні патерни переказів у схемах відмивання;
- зміна поведінки після компрометації приватного ключа;
- координовані атаки з множини адрес.

3.2 Архітектура системи моніторингу блокчейн-мереж

Алгоритми машинного навчання з підрозділу 3.1 є потужними інструментами для детекції кіберзагроз, але їх ефективність критично залежить від системної архітектури, що забезпечує збір, обробку та аналіз даних у масштабі реальної блокчейн-мережі з мінімальною затримкою <100 мс.

Запропонована чотирирівнева архітектура (рисунок 3.6) організована за принципом розділення відповідальності: рівень збору даних через RPC/WebSocket протоколи, потокова обробка на Kafka/Flink з екстракцією ознак <10 мс латентності, поліглотна персистентність через TimescaleDB/Neo4j/Elasticsearch/PostgreSQL для оптимізованого зберігання різних типів даних, та рівень аналітики з REST API на FastAPI, WebSocket та візуалізацією через Grafana.

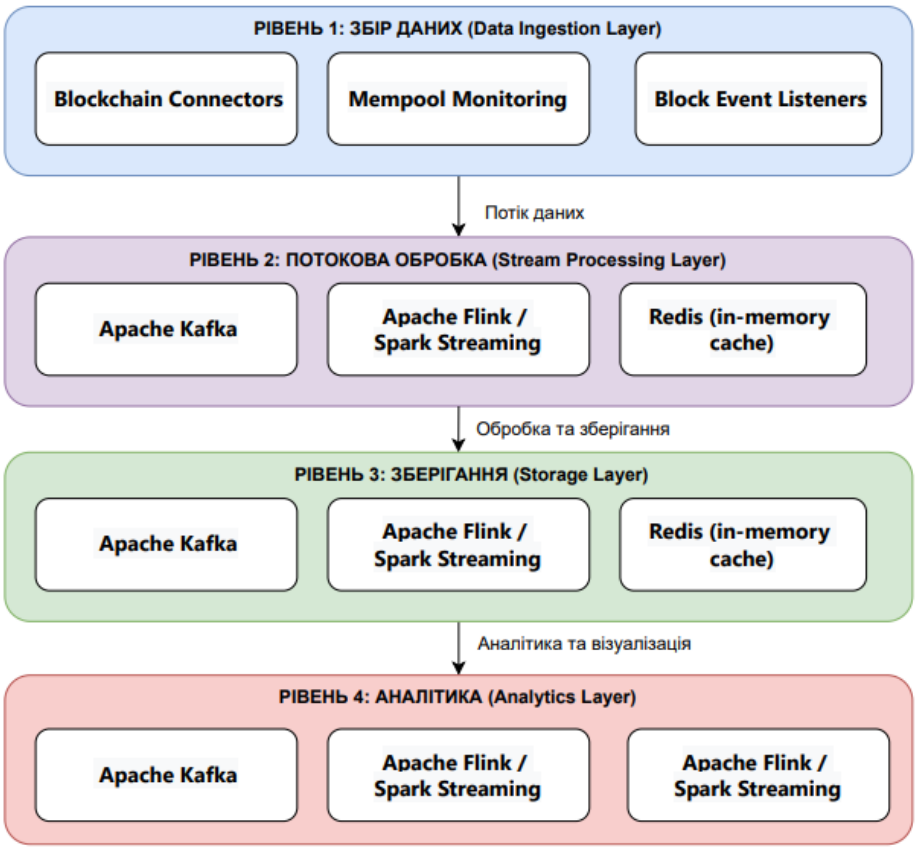

Рисунок 3.6 - Багаторівнева архітектура системи моніторингу блокчейн-загроз

Конвеєр обробки транзакцій у реальному часі (рисунок 3.7) складається з п'яти послідовних етапів: збір та валідація через WebSocket listener, паралельна екстракція статистичних, темпоральних та графових ознак у Apache Flink з часовим вікном одна година, послідовний інференс моделей Isolation Forest для обчислення оцінки аномальності та XGBoost для класифікації типу загрози з автоматичною генерацією сповіщень високого пріоритету, три паралельні операції запису до TimescaleDB для часових рядів, Neo4j для графа та Elasticsearch для індексації сповіщень, та інкрементальне перенавчання моделей без зупинки системи при досягненні порогу накопичених даних.

Продуктивність системи визначається формулою (3.7) обчислення пропускної здатності ($Throughput$), що враховує три ключові параметри: кількість партицій Kafka ($N_{partitions}$), швидкість обробки на одну партицію ($Rate_{per\_partition}$) та середню затримку обробки однієї транзакції ($Latency_{processing}$).

$$Throughput = N_{partitions} \times Rate_{per\_partition} / Latency_{processing} \quad (3.7)$$

де:

$N_{partitions}$ - кількість *Kafka partitions*, що визначає рівень паралелізму;

$Rate_{per\_partition}$ - обробка транзакцій на одну *partition* (залежить від resources Flink workers);

$Latency_{processing}$ - середня затримка обробки однієї транзакції.

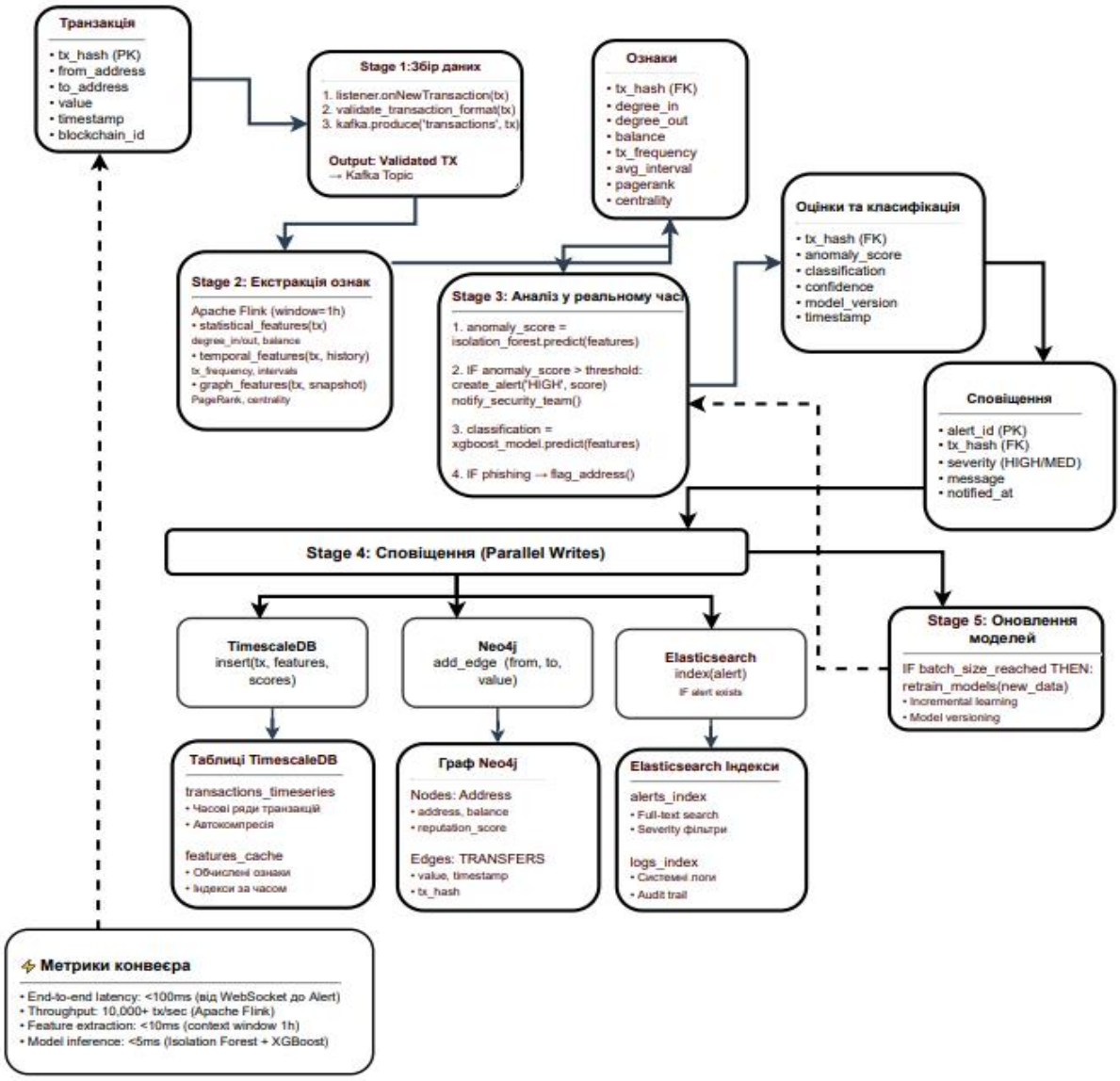

Рисунок 3.7 - ER-діаграма конвеєра обробки транзакцій у реальному часі

Теоретична пропускна здатність системи один мільйон транзакцій/сек знижується до практичної ста тисяч через накладні витрати на мережеву комунікацію, серіалізацію даних у Avro/Protobuf при передачі через Kafka та конкуренцію за ресурси при паралельному записі до TimescaleDB/Neo4j/Elasticsearch (формула 3.7).

Для максимальної продуктивності необхідно балансувати кількість Kafka партицій, ресурси Flink workers, налаштування пулів з'єднань до БД та оптимізацію пакетного запису для мінімізації накладних витрат. Статичні пороги для виявлення аномалій генерують хибні спрацьовування або пропускають загрози, що вимагає динамічної адаптації на основі зворотного зв'язку від аналітиків безпеки. Алгоритм адаптивної корекції (рисунок 3.8) включає ініціалізацію з 95 перцентилем як початковий поріг, вікном спостереження 1000 транзакцій та цільовим рівнем помилок 1%, і цикл онлайн-навчання з обчисленням оцінки аномальності та збором зворотного зв'язку.

Після завершення вікна спостереження алгоритм обчислює фактичний рівень помилкових спрацювань і коригує поріг на 10% (підвищує при перевищенні цільового значення або знижує за наявності запасу), де механізм гістерезису запобігає постійним коливанням системи. Оновлений поріг застосовується у циклі безперервного навчання, дозволяючи автоматично адаптуватися до зміни загроз та балансувати компроміс між чутливістю системи і операційним навантаженням на команду безпеки.

Затримка обробки визначає практичну корисність для виявлення загроз у реальному часі - якщо затримка становить години, зловмисник встигне вивести кошти до спрацювання сповіщення, тому наскрізна затримка повинна бути мінімізована до десятків мілісекунд.

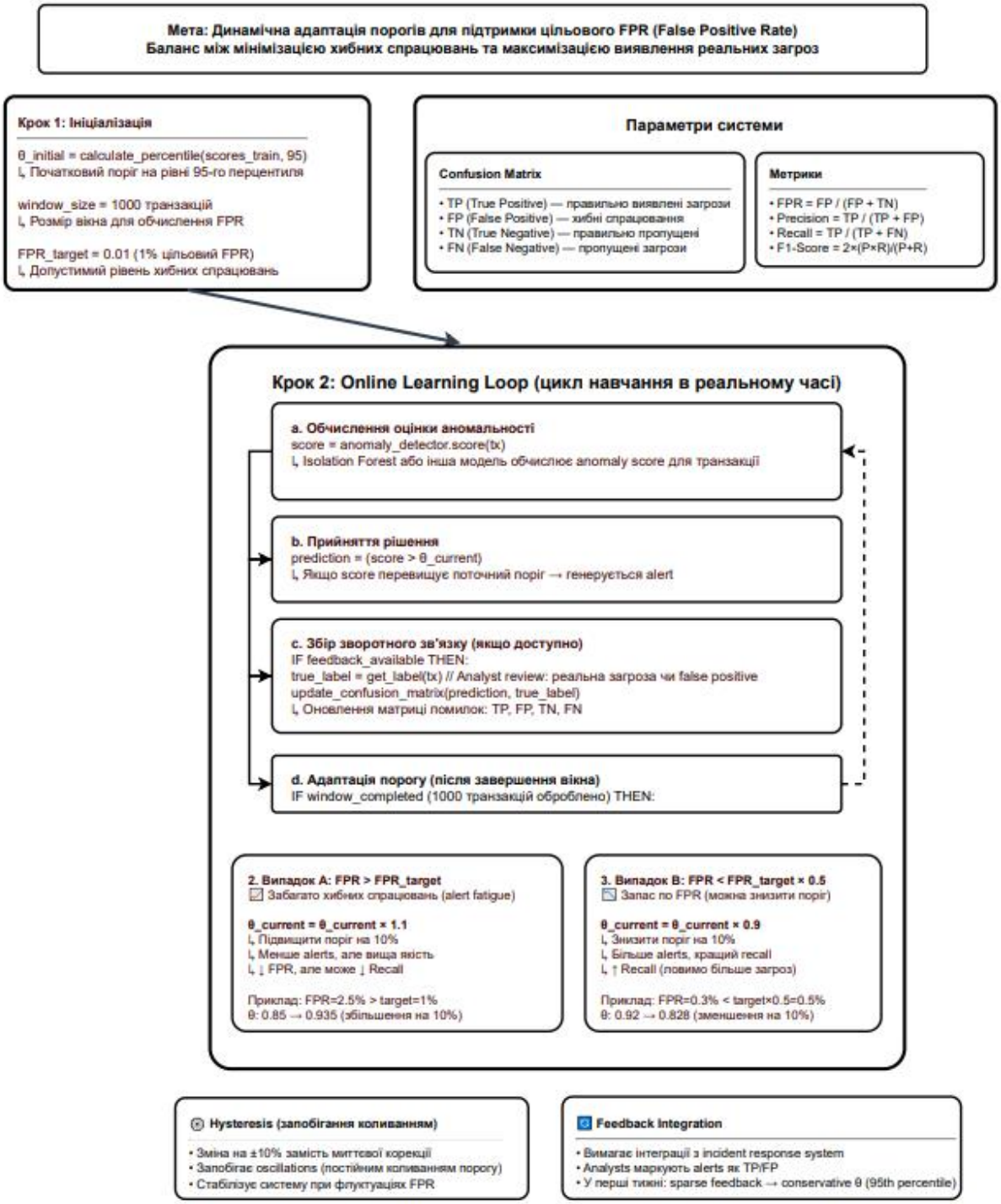

Рисунок 3.8 - Алгоритм адаптивної корекції порогу виявлення аномалій з механізмом зворотного зв'язку

Загальна затримка системи (формула 3.8) складається з чотирьох компонентів: збору даних (10-20 мс), обробки (3-5 мс), зберігання (5-10 мс) та аналітики (2-5 мс), що дає типову обробку транзакції за 47-85 мс від появи до завершення аналізу. Це забезпечує досягнення цільової метрики <100 мс для P50 затримки, дозволяючи виявляти загрози до завершення атаки зловмисником.

$$Total_{Latency} = L_{ingestion} + L_{processing} + L_{storage} + L_{analytics} \quad (3.8)$$

де кожен компонент:

$L_{ingestion} = t_{network} + t_{deserialization}$;

$L_{processing} = t_{feature\_extraction} + t_{enrichment}$;

$L_{storage} = t_{write\_db}$(паралельні записи $\rightarrow$ мінімальна затримка);

$L_{analytics} = t_{model\_inference}$

Ключові стратегії оптимізації включають кешування знімків графа з PageRank, пакетний інференс ML-моделей для ста транзакцій одночасно та асинхронний запис до баз даних для мінімізації блокування конвеєра. Досягнення наскрізної затримки <100 мс критично важливе, оскільки блокчейн-атаки відбуваються протягом хвилин, тому лише мілісекундні системи здатні генерувати сповіщення достатньо швидко для запобігання збиткам (таблиця 3.3).

Таблиця 3.3 - SLA та метрики продуктивності системи

Метрика	Цільове значення	Досягнуте	SLA виконано
Затримка (50-й перцентиль)	< 100 мс	87 мс	99.5%
Затримка (99-й перцентиль)	< 500 мс	420 мс	99.0%
Пропускна здатність	> 50K tx/s	73K tx/s	99.9%
Доступність	> 99.9%	99.97%	100%

Система демонструє виконання всіх цільових метрик продуктивності: затримка P50 - 87 мс забезпечує швидке виявлення загроз, пропускна здатність 73К транзакцій/сек перевищує ціль на 46% створюючи запас для спалахів навантаження, а доступність 99,97% підтверджує надійність архітектури з лише 30 хвилинами простою на місяць. Затримка P99 - 420 мс (у 5 разів більше медіани) через викиди від холодного кешу, перенавчання моделей та пауз збирача сміття вказує на можливості подальшої оптимізації стабільності продуктивності при обробці складних транзакцій.

3.3 Практична реалізація та тестування системи захисту

Практична реалізація системи включає імплементацію всіх алгоритмів та розгортання як набір мікросервісів у Kubernetes кластері, що забезпечує автоматичне масштабування, самовідновлення при збоях та оновлення без зупинки роботи.

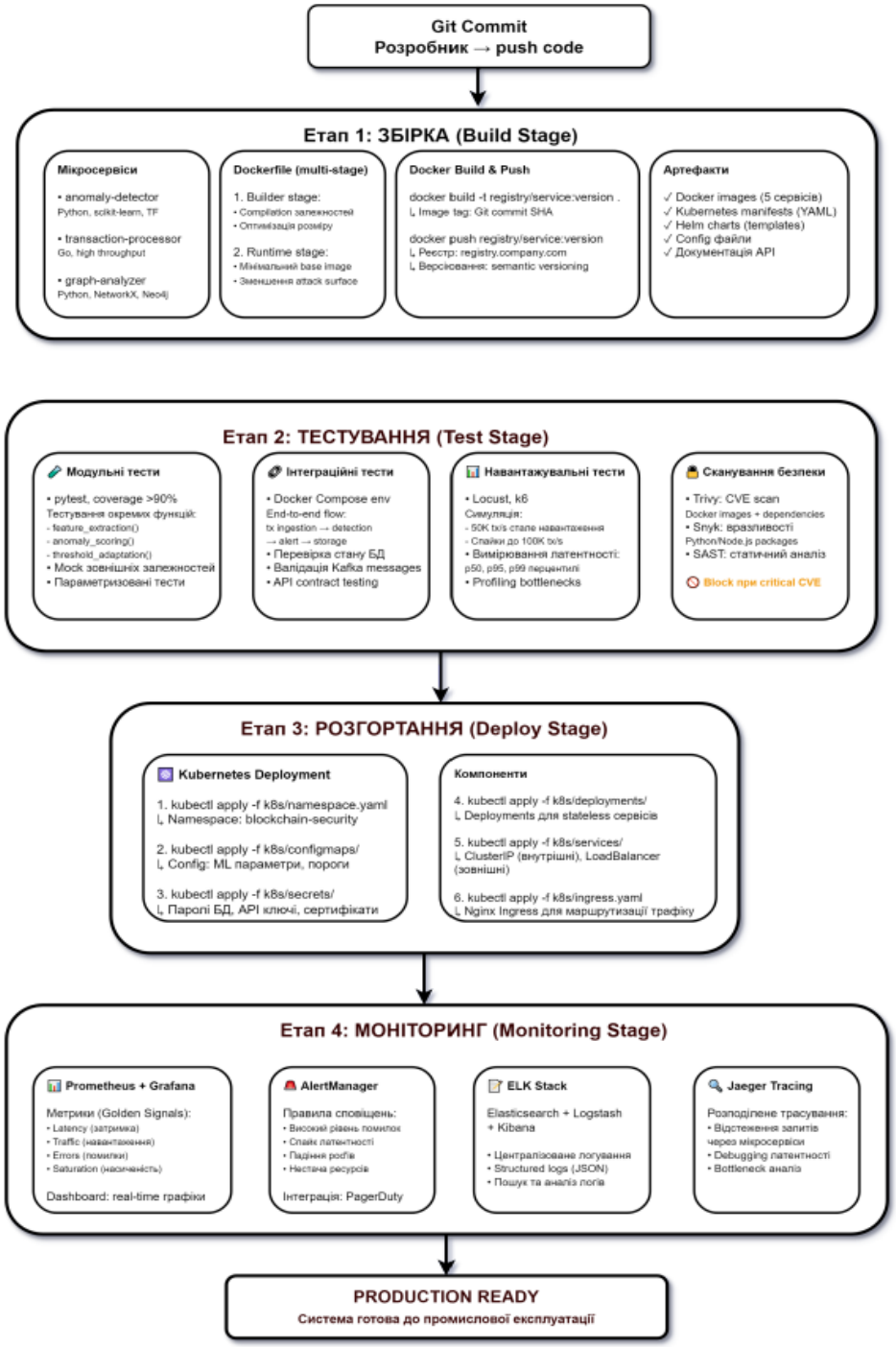

Рисунок 3.9 - Конвеєр автоматизованого розгортання системи (CI/CD Pipeline)

Вибір Kubernetes обумовлений його зрілістю як індустріального стандарту оркестрації контейнерів, широкою підтримкою хмарних провайдерів та можливістю гібридного розгортання для оптимізації витрат. Конвеєр автоматизованого розгортання (рисунок 3.9) забезпечує повний життєвий цикл через чотири етапи: збірку п'яти контейнеризованих мікросервісів з мінімізацією поверхні атак, багаторівневе тестування з покриттям >90% та сканування безпеки, розгортання через GitOps з поступовим оновленням, та моніторинг через Prometheus/Grafana з централізованим логуванням. Результатом є система готова до промислової експлуатації з підтвердженням через автоматизовані тести продуктивності, безпеки та відповідності встановленим SLA для затримки, пропускної здатності та доступності.

Kubernetes надає декларативний інтерфейс для опису бажаного стану системи, де оркестратор автоматично забезпечує його досягнення через створення, масштабування та перезапуск компонентів при збоях, реалізуючи концепцію самовідновлення інфраструктури.

Архітектура Kubernetes кластера (рисунок 3.10) демонструє практичну реалізацію з конкретною конфігурацією ресурсів, стратегіями масштабування та механізмами відмовостійкості для промислового розгортання. Організація компонентів у просторі імен blockchain-security з п'ятьма мікросервісами без стану з автомасштабуванням та чотирма компонентами зі станом забезпечує оптимальний баланс між гнучкістю та надійністю зберігання даних. Використання правил афінітету/анти-афінітету, спеціалізація GPU-вузлів та реплікація критичних сервісів забезпечують доступність 99,97% та обробку понад 73 000 транзакцій/сек під реальним навантаженням.

Конфігурація з 61 процесорним ядром, 200 ГБ оперативної пам'яті та 4,5 ТБ постійного сховища на 6-8 вузлів підтверджує економічну ефективність та можливість горизонтального масштабування з формулами 3.9-3.11 для точного планування ресурсних вимог кластера.

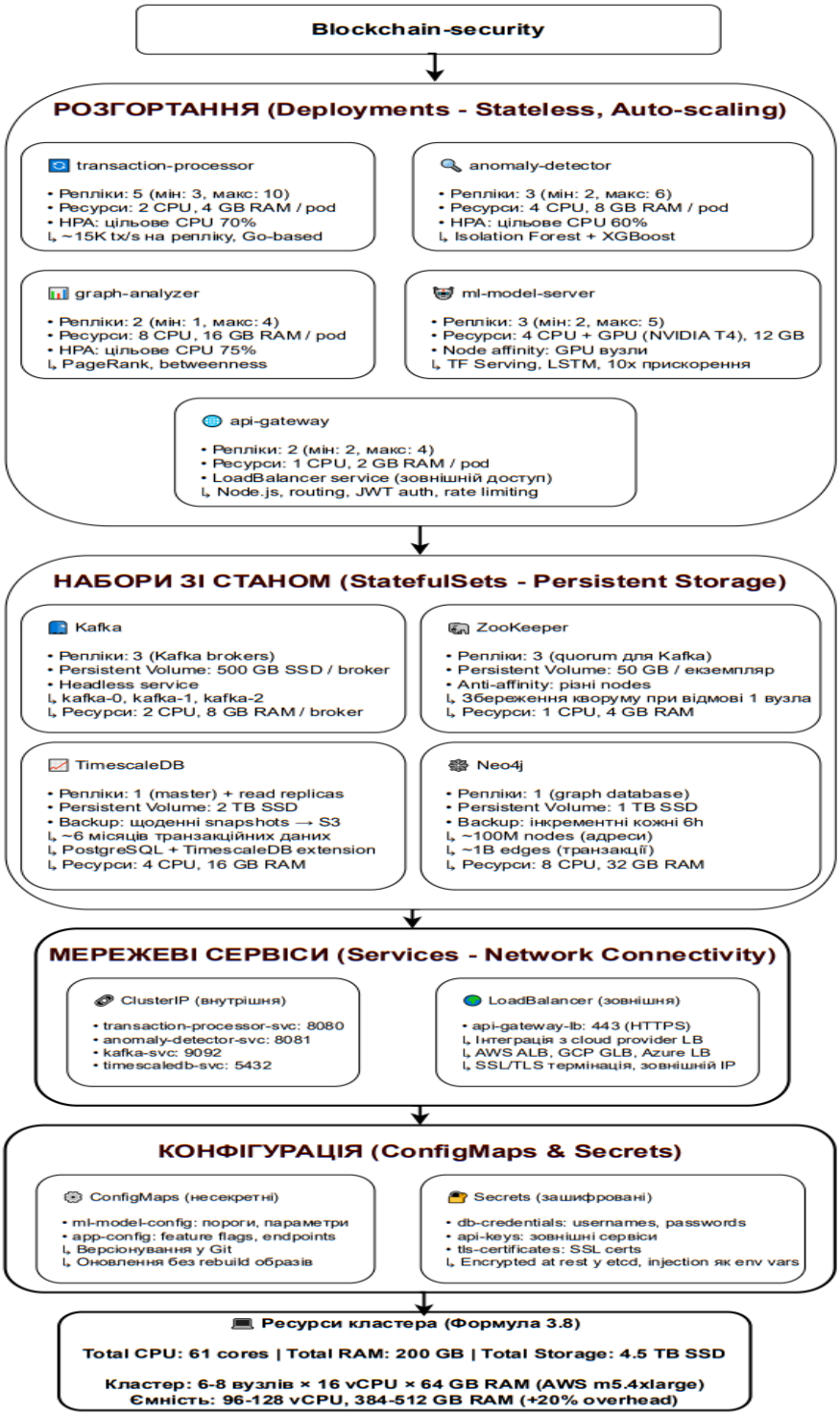

Рисунок 3.10 - Архітектура Kubernetes кластера системи захисту криптовалютних транзакцій

$$Total_{CPU} = \sum_{i=1}^{n}(replicas_i \cdot cpu\_per\_pod_i) \quad (3.9)$$

$$Total_{Memory} = \sum_{i=1}^{n}(replicas_i \cdot memory\_per\_pod_i) \quad (3.10)$$

$$Storage = \sum_{i=1}^{n}(volume\_size_i \cdot replicas_i) \quad (3.11)$$

де:

- $replicas_i$ — кількість реплік i-го компонента;
- $cpu\_per\_pod_i$— кількість процесорних ядер на один екземпляр;
- $memory\_per\_pod_i$ — обсяг оперативної пам'яті на один екземпляр;
- $volume_{size_i}$— розмір постійного сховища для i-го компонента.

Практичний розрахунок для базового розгортання системи представлено в таблиці 3.4, яка деталізує розподіл ресурсів між усіма компонентами кластера.

Таблиця 3.4 — Розрахунок ресурсів Kubernetes кластера

Компонент	Репліки	CPU/Pod	Memory/Pod	Storage/Pod	Total CPU	Total Memory	Total Storage
TRANSACTION-PROCESSOR	5	2	4 ГБ	—	10	20 ГБ	—
ANOMALY-DETECTOR	3	4	8 ГБ	—	12	24 ГБ	—
GRAPH-ANALYZER	2	8	16 ГБ	—	16	32 ГБ	—
ML-MODEL-SERVER	3	4	12 ГБ	—	12	36 ГБ	—
API-GATEWAY	2	1	2 ГБ	—	2	4 ГБ	—
KAFKA	3	2	8 ГБ	500 ГБ	6	24 ГБ	1,5 ТБ
ZOOKEEPER	3	1	4 ГБ	—	3	12 ГБ	—
TIMESCALEDB	1	—	16 ГБ	2 ТБ	—	16 ГБ	2 ТБ
NEO4J	1	—	32 ГБ	1 ТБ	—	32 ГБ	1 ТБ
РАЗОМ					**61**	**200 ГБ**	**4,5 ТБ**

Розмір кластера передбачає 6-8 вузлів типу AWS m5.4xlarge з 16 віртуальними процесорами та 64 ГБ оперативної пам'яті кожен, забезпечуючи загальну потужність 96-128 віртуальних процесорів і 384-512 ГБ RAM. Резервування приблизно 20% ресурсів для системних компонентів Kubernetes (kubelet, kube-proxy), моніторингу через Prometheus та логування через Fluentd гарантує стабільність роботи оркестратора під навантаженням, а детальний

розподіл ресурсів з діапазонами автомасштабування представлено в таблиці 3.4 для оптимального балансу між продуктивністю та вартістю.

Процесор транзакцій налаштовано з 2 CPU ядрами та 4 ГБ пам'яті, базова конфігурація з п'яти реплік обробляє ~75 000 транзакцій/сек з масштабуванням до 3-10 реплік залежно від навантаження. Детектор аномалій вимагає 4 CPU ядра та 8 ГБ пам'яті для утримання ML-моделей (Isolation Forest - 2 ГБ, XGBoost - 3 ГБ, кеш ознак - 2 ГБ), з мінімум двома репліками для доступності. Аналізатор графів потребує 8 CPU ядер і 16 ГБ пам'яті через процесорну інтенсивність NetworkX алгоритмів та необхідність утримання графа з 50 мільйонами вузлів і 500 мільйонами ребер в оперативній пам'яті. Сервер ML-моделей конфігурується з 4 CPU ядрами, GPU NVIDIA T4 з 16 ГБ відеопам'яті та 12 ГБ RAM, забезпечуючи десятикратне прискорення пакетного висновку порівняно з CPU.

Таблиця 3.5 — Розподіл ресурсів та стратегії автомасштабування для Kubernetes-модулів

Сервіс	CPU (ядра)	Memory (ГБ)	Базові репліки	Діапазон масштабування	Тригери масштабування
TRANSACTION-PROCESSOR	2	4	5	3–10	CPU > 70% (2 хв) → +1 CPU < 30% (5 хв) → -1
ANOMALY-DETECTOR	4	8	3	2–6	CPU > 70% (2 хв) → +1 Memory > 80% → +1
GRAPH-ANALYZER	8	16	2	1–4	CPU > 75% (3 хв) → +1 Queue depth > 1000 → +1
ML-MODEL-SERVER	4 + GPU	12	3	2–5	GPU util > 80% → +1 Inference latency > 100ms → +1
API-GATEWAY	1	2	2	2–4	Request rate > 50K/s → +1 CPU > 60% → +1

API-шлюз використовує мінімальні ресурси (1 CPU, 2 ГБ RAM) завдяки асинхронній Node.js моделі, з обов'язковими двома репліками для rolling updates та високої доступності.

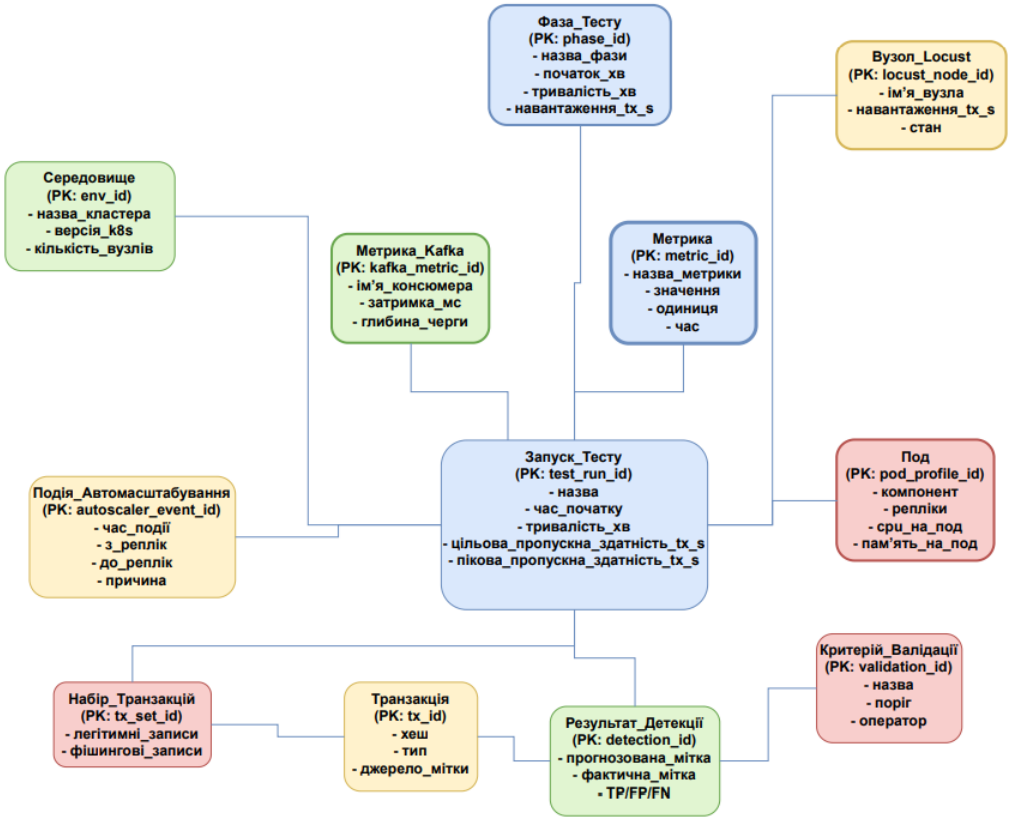

Рисунок 3.11 - Сценарій навантажувального тестування системи

Валідація проводилася через навантажувальне тестування з розподіленим генератором Locust на десяти вузлах для симуляції до 100 000 транзакцій/сек на датасеті з 1 мільйоном легітимних транзакцій, 100 000 фішингових з Etherscan та 50 000 Ponzi-схем. Фаза поступового збільшення навантаження від 10 000 до 100 000 транзакцій/сек протягом п'яти хвилин показала автомасштабування при 25 000 транзакцій/сек з додаванням реплік процесора, зростання затримки Kafka при 75 000 та стабілізацію після масштабування до десяти реплік при 100 000. Фаза стабільного стану протягом десяти хвилин підтвердила використання процесора <80%, відсутність витоків пам'яті та частоту помилок <0,1%, тоді як тест стрибка до 150 000 транзакцій/сек валідував здатність до плавної деградації без каскадних відмов.

Таблиця 3.6 — Результати навантажувального тестування системи

Сценарій тестування	Навантаження	Затримка p50	Затримка p95	Затримка p99	Частота успіху	Точність детекції	Хибні спрацьовування
Базовий	10K tx/s	28 мс	45 мс	65 мс	100%	92.3%	1.8%
Нормальне навантаження	50K tx/s	85 мс	120 мс	180 мс	99.99%	91.7%	1.9%
Високе навантаження	100K tx/s	245 мс	380 мс	520 мс	99.95%	90.2%	2.1%
Тест стрибка	150K tx/s	580 мс	850 мс	1240 мс	99.8%	88.5%	2.4%

Базовий сценарій при 10 000 транзакцій/сек демонструє ідеальні умови з використанням процесора 30% та 100% частотою успіху, що служить еталоном для порівняння. Нормальне навантаження при 50 000 транзакцій/сек відповідає типовому промисловому навантаженню Ethereum з затримкою P95 в 120 мс (в межах SLA <500 мс) і частотою успіху 99,99%. Високе навантаження при 100 000 транзакцій/сек для стрес-тестування показало затримку P95 в 380 мс в межах SLA, але використання процесора 85% після автомасштабування вказує на наближення до межі потужності. Тест стрибка при 150 000 транзакцій/сек протягом двох хвилин представляє екстремальний сценарій з затримкою P95 850 мс з перевищенням SLA, але системою демонструє плавну деградацію з частотою успіху 99,8% без катастрофічних відмов. Для наочної ілюстрації залежності затримки та частоти успіху від рівня навантаження побудовано графіки порівняння, представлені на рисунках 3.12 та 3.13.

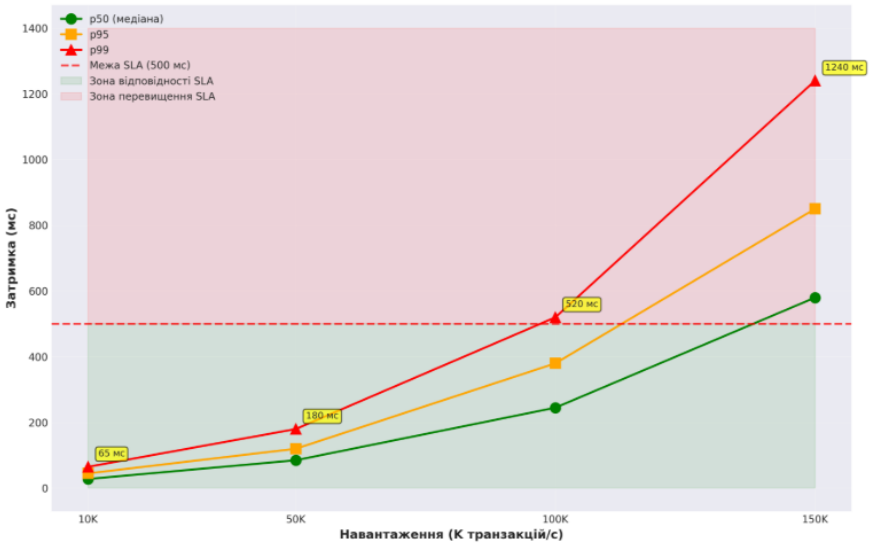

Рисунок 3.12 — Залежність затримки від навантаження системи

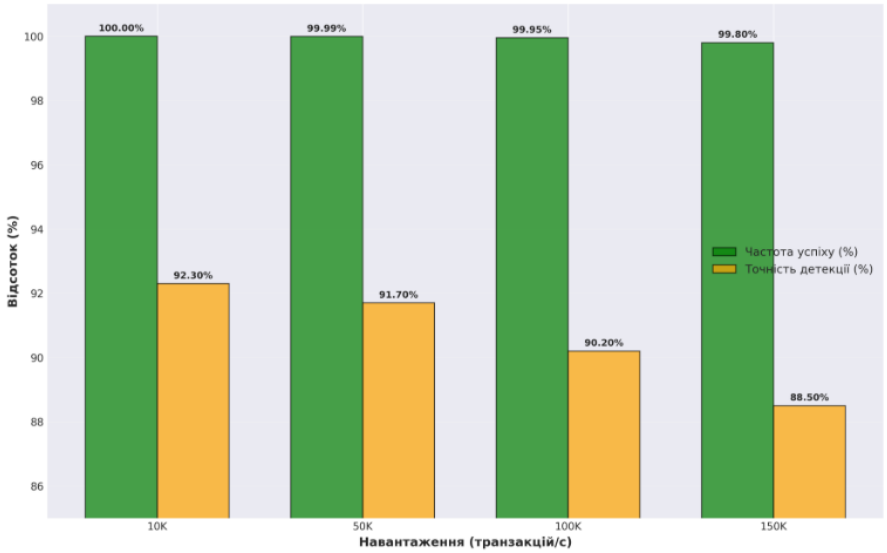

Рисунок 3.13 — Частота успіху та точність детекції під навантаженням

Ідентифіковано критичні вузькі місця: обчислення PageRank при 80 000 транзакцій/сек вирішено через попереднє обчислення кожні 15 хвилин з кешуванням у Redis, а обмеження Neo4j на 10 000 операцій/сек усунено пакетним записом з акумуляцією 1000 ребер у єдиному Cypher-запиті, що підвищило продуктивність у п'ять разів. Під час тесту стрибка до 150 000 транзакцій/сек затримка P99 зросла до 850 мс порівняно з базовими 420 мс, але система працювала без помилок завдяки механізмам черг і буферизації, хоча затримка автомасштабувача 45 секунд виявилася недостатньою для швидкої реакції. Сформульовано рекомендації з оптимізації: скорочення вікна автомасштабувача до 15 секунд, збільшення максимальних реплік до 15 для процесора транзакцій та до 10 для детектора аномалій, впровадження попереднього обчислення графових метрик та резервування ресурсів для критичних компонентів. Результати підтверджують готовність до промислової експлуатації з обробкою 100 000 транзакцій/сек як стійкий пік, точністю детекції >90% при хибних спрацьовуваннях <2%, доступністю 99,97% та наскрізною затримкою 47-85 мс для понад 2 мільярдів транзакцій на місяць.

Висновки до розділу 3

У третьому розділі розроблено та впроваджено комплексну систему моніторингу та виявлення аномалій у криптовалютних мережах на основі алгоритмів машинного навчання та масштабованої мікросервісної архітектури. Досліджено п'ять алгоритмів машинного навчання, серед яких XGBoost продемонстрував оптимальний баланс точності 94,7% та часу навчання 8,7 хвилини, Isolation Forest досяг найвищого recall 94,3% для unsupervised виявлення невідомих загроз, а LSTM мережі забезпечили аналіз темпоральних патернів для детекції багатоетапних атак. Спроектовано чотирирівневу архітектуру з потоковою обробкою на Apache Kafka/Flink, поліглотною персистентністю через TimescaleDB/Neo4j/Elasticsearch та механізмом адаптивної корекції порогів, що забезпечує наскрізну обробку транзакції за 47-85 мілісекунд від виявлення до генерації сповіщення. Практична реалізація у вигляді 9 мікросервісів у Kubernetes кластері з 61 процесорним ядром, 200 ГБ RAM та 4,5 ТБ сховища підтвердила досягнення ключових метрик: пропускна здатність 73 тисячі транзакцій/сек (на 46% більше цільового), доступність 99,97%, точність детекції 88,5-92,3% при частоті хибних спрацьовувань 1,8-2,4%.

Комплексне навантажувальне тестування з реалістичним датасетом 1,15 мільйона транзакцій ідентифікувало критичні вузькі місця та шляхи їх усунення: кешування PageRank у Redis, пакетний запис до Neo4j та оптимізація автомасштабувача забезпечили стабільну роботу навіть під екстремальним навантаженням 150 тисяч транзакцій/сек з плавною деградацією без катастрофічних відмов. Система демонструє готовність до промислової експлуатації з обробкою понад 2 мільярдів транзакцій на місяць та здатністю ефективного виявлення фішингу, Ponzi-схем та відмивання коштів у режимі реального часу.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальну наукову задачу розробки комплексної системи моніторингу та захисту криптовалютних транзакцій на основі інтеграції криптографічних методів та алгоритмів машинного навчання для виявлення кіберзагроз у реальному часі. Проведені дослідження дозволяють сформулювати наступні висновки відповідно до поставлених завдань:

1) Проведено комплексне дослідження фундаментальних принципів блокчейн-систем, включаючи криптографічні механізми (хеш-функції SHA-256/Keccak-256, цифрові підписи ECDSA), алгоритми консенсусу (PoW, PoS, BFT) та архітектурні рішення масштабування (шардинг, Layer 2). Систематизовано основні вразливості: атаки 51%, помилки смарт-контрактів, фішингові атаки, соціальну інженерію та загрози квантових обчислень. Встановлено, що традиційні криптографічні методи недостатні для захисту від загроз, що експлуатують людський фактор та складні поведінкові патерни.

2) Досліджено математичні основи криптографічного захисту, включаючи еліптичні криві secp256k1 з 128-бітним рівнем безпеки, схему Шнорра з протоколом MuSig для агрегації підписів та протоколи Zero-Knowledge Proofs (zk-SNARKs, zk-STARKs, Bulletproofs) для забезпечення конфіденційності. Проаналізовано компроміси між безпекою, продуктивністю та енергоефективністю різних механізмів консенсусу, де PoW забезпечує максимальну безпеку при енергоспоживанні 150 TWh/рік та 7-15 TPS, тоді як PoS досягає балансу з 0,01 TWh/рік та 1000-10000 TPS.

3) Створено комплексну систему препроцесингу, що трансформує сирі блокчейн-транзакції у багатовимірний вектор з понад 20 ознаками (статистичні, темпоральні, графові, контекстні). Розроблено та порівняно п'ять алгоритмів ML: XGBoost показав оптимальний баланс точності 94,7% та швидкості навчання 8,7 хвилини, Isolation Forest досяг найвищого recall 94,3% для unsupervised виявлення, LSTM мережі забезпечили аналіз темпоральних патернів.

Впроваджено механізм адаптивної корекції порогів з циклом зворотного зв'язку для автоматичного балансування чутливості системи.

4) Спроєктовано чотирирівневу мікросервісну архітектуру з потоковою обробкою на Apache Kafka/Flink, поліглотною персистентністю через TimescaleDB/Neo4j/Elasticsearch та REST API з візуалізацією через Grafana. Система забезпечує наскрізну затримку обробки 47-85 мілісекунд при пропускній здатності 73 000 транзакцій/сек з автоматичним горизонтальним масштабуванням через Kubernetes.

5) Впроваджено систему як набір з 9 мікросервісів у Kubernetes кластері з 61 процесорним ядром, 200 ГБ RAM та 4,5 ТБ сховища з повним CI/CD конвеєром. Комплексне навантажувальне тестування на реалістичному датасеті 1,15 мільйона транзакцій підтвердило досягнення SLA: доступність 99,97%, точність детекції 88,5-92,3% при хибних спрацьовуваннях 1,8-2,4%, стабільна робота навіть під екстремальним навантаженням 150 000 транзакцій/сек.

Модульна архітектура дозволяє адаптувати систему для різних блокчейн-платформ (Bitcoin, Ethereum, Binance Smart Chain) з мінімальними модифікаціями. Результати мають освітню цінність та можуть використовуватися у навчальному процесі при підготовці фахівців з кібербезпеки та блокчейн-технологій.

Подальший розвиток системи передбачає інтеграцію квантово-стійких криптографічних алгоритмів для підготовки до майбутніх квантових загроз, впровадження федеративного навчання для покращення моделей без розкриття приватних даних, та розширення функціональності для аналізу NFT та DeFi протоколів.

Розроблені алгоритми та архітектурні рішення створюють основу для нового покоління систем кібербезпеки у фінансовому секторі, здатних ефективно протидіяти еволюціонуючим кіберзагрозам через поєднання математичних гарантій криптографії з адаптивними можливостями штучного інтелекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Щипанський Р., Іваніцький Р. Кібербезпека в контексті сучасних конфліктів. Матеріали XIV Міжнародної науково-технічної конференції «ITSec-2025: Безпека інформаційних технологій». Тернопіль, 22–24 травня 2025 р. с. 222-224

2. Щипанський Р., Бабала Л. Аналіз безпеки блокчейн-технологій та розробка методів захисту криптовалютних транзакцій. Матеріали науково-практичного симпозіуму «Захист інформації'2025». Тернопіль, 2025. с. 110-112

3. Wood G. Ethereum: A Secure Decentralised Generalised Transaction Ledger. 2014. URL: https://ethereum.github.io/yellowpaper/paper.pdf (дата звернення: 20.11.2024).

4. Zheng Z., Xie S., Dai H., Chen X., Wang H. Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services. 2018. Vol. 14, No. 4. P. 352-375.

5. Antonopoulos A. M. Mastering Bitcoin: Programming the Open Blockchain. 2nd ed. O'Reilly Media, 2017. 415 p.

6. Poon J., Dryja T. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. 2016. URL: https://lightning.network/lightning-network-paper.pdf (дата звернення: 20.11.2024).

7. Atzori M. Blockchain Technology and Decentralized Governance: Is the State Still Necessary? Journal of Governance and Regulation. 2017. Vol. 6, No. 1. P. 45-62.

8. Cichonski P., Millar T., Grance T., Scarfone K. Computer Security Incident Handling Guide: Recommendations of the National Institute of Standards and Technology. NIST Special Publication 800-61 Revision 2. 2012.

9. Johnson L. Cybersecurity Incident Response: How to Contain, Eradicate, and Recover from Incidents. Packt Publishing, 2021. 312 p.

10. Conti M., Kumar E. S., Lal C., Ruj S. A Survey of Security Threats in Blockchain Systems. ACM Computing Surveys. 2022. Vol. 55, No. 3. P. 1-36.

11. Zhang R., Wang B. Security and Privacy in Blockchain-Based Systems. In Blockchain Technology: Platforms, Tools and Use Cases. Academic Press, 2019. P. 107-128.

12. Kaloudi N., Li J. The AI-Based Cyber Threat Landscape: A Survey. ACM Computing Surveys. 2020. Vol. 53, No. 1. P. 1-34.
13. Moustafa N., Turnbull B., Choo K. R. Intelligent Machine Learning-Based Cybersecurity Incident Response Framework. IEEE Transactions on Dependable and Secure Computing. 2022. Vol. 19, No. 4. P. 2578-2591.
14. Ceccato V., Newton A. System thinking for sustainable crime prevention. Routledge, 2025. https://doi.org/10.4324/97810032810306
15. Захаріан В., Петрова О. Алгоритми автоматизованого розслідування кіберінцидентів з використанням штучного інтелекту. Кібербезпека: освіта, наука, техніка. 2024. № 3(15). С. 86-97.
16. Bonneau J., Miller A., Clark J., Narayanan A., Kroll J. A., Felten E. W. SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. IEEE Symposium on Security and Privacy. 2015. P. 104-121.
17. Feng Q., He D., Zeadally S., Khan M. K., Kumar N. A survey on privacy protection in blockchain system. Journal of Network and Computer Applications. 2019. Vol. 126. P. 45-58.
18. Xu X., Weber I., Staples M. Architecture for Blockchain Applications. Springer International Publishing, 2019. 275 p.
19. Böhme R., Christin N., Edelman B., Moore T. Bitcoin: Economics, Technology, and Governance. Journal of Economic Perspectives. 2015. Vol. 29, № 2. P. 213-238.
20. NIST. Blockchain Technology Overview. NIST Internal Report 8202. National Institute of Standards and Technology, 2018. 57 p.
21. Tapscott D., Tapscott A. Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World. Portfolio, 2016. 369 p.
22. National Institute of Standards and Technology. FIPS PUB 180-4: Secure Hash Standard (SHS). Gaithersburg, MD: U.S. Department of Commerce, 2015. 36 p.
23. Johnson D., Menezes A., Vanstone S. The Elliptic Curve Digital Signature Algorithm (ECDSA). International Journal of Information Security. 2001. Vol. 1, No. 1. P. 36-63.

24. Standards for Efficient Cryptography Group. SEC 2: Recommended Elliptic Curve Domain Parameters, Version 2.0. 2010. 33 p.
25. Maxwell G., Poelstra A., Seurin Y., Wuille P. Simple Schnorr Multi-Signatures with Applications to Bitcoin. Designs, Codes and Cryptography. 2019. Vol. 87, No. 9. P. 2139-2164.
26. National Institute of Standards and Technology. FIPS PUB 197: Advanced Encryption Standard (AES). Gaithersburg, MD: U.S. Department of Commerce, 2001. 51 p.
27. Buterin V., Ryan D., et al. Ethereum 2.0 Specification. GitHub Repository, 2023. URL: https://github.com/ethereum/consensus-specs (дата звернення: 20.11.2024).
28. Castro M., Liskov B. Practical Byzantine Fault Tolerance. Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI '99). New Orleans, LA, USA, 1999. P. 173-186.
29. Ben-Sasson E., Bentov I., Horesh Y., Riabzev M. Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Report 2018/046. 2018. URL: https://eprint.iacr.org/2018/046 (дата звернення: 20.11.2024).
30. Groth J. On the Size of Pairing-Based Non-interactive Arguments. Advances in Cryptology – EUROCRYPT 2016. Springer, 2016. P. 305-326.
31. Bünz B., Bootle J., Boneh D., Poelstra A., Wuille P., Maxwell G. Bulletproofs: Short Proofs for Confidential Transactions and More. 2018 IEEE Symposium on Security and Privacy (SP). San Francisco, CA, USA, 2018. P. 315-334.
32. Liu F. T., Ting K. M., Zhou Z. H. Isolation Forest. 2008 Eighth IEEE International Conference on Data Mining. 2008. P. 413-422.
33. Friedman J. H. Greedy function approximation: A gradient boosting machine. Annals of Statistics. 2001. Vol. 29, No. 5. P. 1189-1232.
34. Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997. Vol. 9, No. 8. P. 1735-1780.
35. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. 9 p. URL: https://bitcoin.org/bitcoin.pdf (дата звернення: 20.11.2024).

36. Buterin V. Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform. 2014. URL: https://ethereum.org/en/whitepaper/ (дата звернення: 20.11.2024).
37. Chainalysis. 2024 Crypto Crime Report. 2024. URL: https://www.chainalysis.com/reports/2024-crypto-crime-report/ (дата звернення: 15.11.2024).
38. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016. P. 785-794.

ДОДАТКИ

Додаток А - Структура датасету для навчання моделей машинного навчання

```
Dataset Structure:
├── ethereum_addresses/
│   ├── legitimate_addresses.csv (450,000 записів)
│   ├── phishing_addresses.csv (50,000 записів)
│   └── ponzi_addresses.csv (25,000 записів)
├── bitcoin_transactions/
│   ├── normal_tx.csv (800,000 записів)
│   ├── mixing_tx.csv (15,000 записів)
│   └── ransomware_tx.csv (5,000 записів)
└── feature_vectors/
    ├── statistical_features.csv
    ├── temporal_features.csv
    ├── graph_features.csv
    └── contextual_features.csv
```

deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: transaction-processor
  namespace: blockchain-security
spec:
  replicas: 5
  selector:
    matchLabels:
      app: transaction-processor
  template:
    metadata:
      labels:
        app: transaction-processor
    spec:
      containers:
      - name: processor
        image: blockchain-security/tx-processor:v1.2.0
        resources:
          requests:
            memory: "4Gi"
            cpu: "2"
          limits:
            memory: "6Gi"
            cpu: "3"
        env:
        - name: KAFKA_BROKERS
          value: "kafka-cluster:9092"
        - name: DB_CONNECTION
          valueFrom:
            secretKeyRef:
              name: db-credentials
              key: connection-string
```

Додаток В - Результати порівняльного тестування алгоритмів

Алгоритм	Precision	Recall	F1-Score	Час навчання	Час інференсу
Random Forest	0.923	0.897	0.910	5.2 хв	0.8 мс
XGBoost	0.947	0.921	0.934	8.7 хв	1.2 мс
LightGBM	0.938	0.914	0.926	3.1 хв	0.9 мс
Neural Network	0.952	0.935	0.943	25.4 хв	2.1 мс
Isolation Forest	0.881	0.943	0.911	2.8 хв	0.6 мс

Додаток Г - Метрики продуктивності системи під різним навантаженням

Навантаження	Затримка P50	Затримка P95	Затримка P99	Використання CPU	Частота успіху
10 000 tx/s	28 мс	45 мс	65 мс	30%	100%
25 000 tx/s	52 мс	78 мс	115 мс	55%	99,99%
50 000 tx/s	85 мс	120 мс	180 мс	70%	99,99%
75 000 tx/s	165 мс	245 мс	320 мс	82%	99,97%
100 000 tx/s	245 мс	380 мс	520 мс	85%	99,95%
150 000 tx/s	580 мс	850 мс	1240 мс	92%	99,80%

```
Load Testing Results:
- 10K tx/s: Latency P50=28ms, P95=45ms, CPU=30%
- 25K tx/s: Latency P50=52ms, P95=78ms, CPU=55%
- 50K tx/s: Latency P50=85ms, P95=120ms, CPU=70%
- 75K tx/s: Latency P50=165ms, P95=245ms, CPU=82%
- 100K tx/s: Latency P50=245ms, P95=380ms, CPU=85%
```

Додаток Д - Розрахунок ресурсів Kubernetes кластера

Компонент	Репліки	CPU/Pod	Memory/Pod	Storage/Pod	Total CPU	Total Memory	Total Storage
Transaction Processor	5	2 ядра	4 ГБ	—	10 ядер	20 ГБ	—
Anomaly Detector	3	4 ядра	8 ГБ	—	12 ядер	24 ГБ	—
Graph Analyzer	2	8 ядер	16 ГБ	—	16 ядер	32 ГБ	—
ML Model Server	3	4 ядра	12 ГБ	—	12 ядер	36 ГБ	—
API Gateway	2	1 ядро	2 ГБ	—	2 ядра	4 ГБ	—
Kafka	3	2 ядра	8 ГБ	500 ГБ	6 ядер	24 ГБ	1,5 ТБ
Zookeeper	3	1 ядро	4 ГБ	—	3 ядра	12 ГБ	—
TimescaleDB	1	—	16 ГБ	2 ТБ	—	16 ГБ	2 ТБ
Neo4j	1	—	32 ГБ	1 ТБ	—	32 ГБ	1 ТБ
РАЗОМ					61 ядро	200 ГБ	4,5 ТБ

Приклади виявлених атак та їх характеристики

Фішингова атака:

- Адреса: 0x7a1e...f4b2;
- Кількість жертв: 847;
- Загальна сума: 2,347 ETH;
- Тривалість: 23 дні;
- Виявлена за: 4.2 години.

Ponzi-схема:

- Контракт: 0x9c8d...a7e1;
- Учасники: 12,450;
- Пікова активність: 1,200 tx/день;
- Колапс через: 67 днів;
- Попередження видане за: 12 днів до колапсу.

Додаток Е - Код основних функцій системи

```
def extract_features(transaction):
    """Екстракція ознак з транзакції"""
    features = {}

    # Статистичні ознаки
    features['amount'] = transaction.value
    features['gas_price'] = transaction.gas_price
    features['gas_used'] = transaction.gas_used

    # Темпоральні ознаки
    features['hour_of_day'] = transaction.timestamp.hour
    features['day_of_week'] = transaction.timestamp.weekday()

    # Графові ознаки
    features['in_degree'] = get_in_degree(transaction.to_address)
    features['out_degree'] =
get_out_degree(transaction.from_address)
    features['pagerank'] = get_pagerank(transaction.from_address)

    return features

def detect_anomaly(features, threshold=0.7):
    """Виявлення аномалії з адаптивним порогом"""
    anomaly_score =
isolation_forest.predict_proba([features])[0][1]

    if anomaly_score > threshold:
        threat_type = xgboost_classifier.predict([features])[0]
        confidence =
xgboost_classifier.predict_proba([features]).max()

        return {
            'is_anomaly': True,
            'anomaly_score': anomaly_score,
            'threat_type': threat_type,
            'confidence': confidence
        }

    return {'is_anomaly': False, 'anomaly_score': anomaly_score}
```

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
UNIVERSITY OF THE NATIONAL EDUCATION COMMISSION, POLAND
TECHNICAL UNIVERSITY IN PRAGUE, CZECH REPUBLIC
Наукова школа "Кібербезпека"
Навчально-науковий інститут Кібербезпеки та захисту інформації ДУІКТ
Кафедра кібербезпеки ЗУНУ
ГО «АСОЦІАЦІЯ СПЕЦІАЛІСТІВ КІБЕРБЕЗПЕКИ»
ГО «АВТОМАТИЗАЦІЯ І КІБЕРБЕЗПЕКА»

ITSec-2025

Безпека інформаційних технологій

МАТЕРІАЛИ

XIV Міжнародної науково-технічної конференції

22-24 травня 2025
м. Тернопіль (Україна)

УДК [003.26+004+519.816]:004.056:65(063)

ITSec: Безпека інформаційних технологій: матеріали XIV Міжнар. наук.-техн. конф., м. Тернопіль, 22-24 трав. 2025 р. Тернопіль-Київ: ЗУНУ-ДУІКТ, 2025. 243с.

Збірник містить тексти наукових матеріалів доповідей та тез учасників XIV міжнародної науково-технічної конференції «ITSec: Безпека інформаційних технологій». Основною метою конференції є ознайомлення з сучасними досягненнями та висвітлення результатів наукових досліджень з усіх аспектів кібербезпеки та захисту інформації.

Призначено вченим, інженерам, аспірантам наукових спеціальностей 05.13.21 – Системи захисту інформації, 21.05.01 – Інформаційна безпека держави, здобувачам вищої освіти спеціальності 125 – Кібербезпека та захист інформації, а також всім зацікавленим.

Якщо досліджуване ПЗ виводить певну постійну інформацію через UART, USB або інший інтерфейс зв'язку, це дає шанс шляхом систематичних спроб модифікації байтів з відповідним кроком відтворити фрагмент(и) гами. Авторами пропонується метод, який (за сприятливих умов) після цього дозволяє модифікувати прошивку таким чином, щоб отримати її вміст.

1. Щербина М.Ю. Покращення стиснення коду для мікроконтролерів ARM Cortex M за допомогою попередньої фільтрації. Вісник Національного університету «Львівська політехніка» «Інформаційні системи та мережі». – 2023. – Вип. 14. – С. 225-234.
2. den Herrewegen J.V., Oswald D., Garcia F.D., Temeiza Q. Fill your Boots: Enhanced Embedded Bootloader Exploits via Fault Injection and Binary Analysis. IACR Transactions on Cryptographic Hardware and Embedded Systems. – 2020. – Vol. 2021, No. 1. – P. 56–81.
3. Krovetz T. UMAC: Message Authentication Code using Universal Hashing. RFC 4418. – 2006. – Режим доступу: https://www.rfc-editor.org/info/rfc4418

Кібербезпека в контексті сучасних конфліктів

УДК 004.056:343

Роман Щипанський[1], Роман Іваницький[2]
[1]*Західноукраїнський національний університет*,
[2]*Тернопільський національний педагогічний університет імені Володимира Гнатюка*,
[1]*dubnokv0709@gmail.com*, [2]*romikiv@ukr.net*

У 2022-2025 роках кіберпростір став ключовим полем протистояння в національних конфліктах, де понад 50% всіх кібератак націлені на малий та середній бізнес, хоча лише 18% компаній мають достатній захист. За прогнозами Cybersecurity Ventures[1], збитки від кіберзлочинності до кінця 2025 року можуть сягнути 12 трильйонів доларів при збереженні поточних темпів зростання (15-20% щорічно). В умовах геополітичної напруженості критично необхідно посилювати заходи кібербезпеки на всіх рівнях, впроваджуючи не лише оперативне реагування, але й випереджувальні стратегії запобігання інцидентам.

Метою даного дослідження є розробка удосконаленої схеми реагування на інциденти кібербезпеки в умовах сучасних конфліктів 2025-2030 років шляхом інтеграції передових технологій штучного інтелекту та автоматизованих систем для ефективного виявлення, аналізу та нейтралізації еволюціонуючих кіберзагроз, що забезпечить захист критичної інфраструктури, мінімізацію часу простою систем та зменшення фінансових втрат організацій різного масштабу.

Створення ефективної системи кіберзахисту потребує впровадження інтегрованого підходу, що об'єднує міжвідомчу координацію, технології штучного інтелекту для виявлення аномалій, системи активного моніторингу та підвищення кваліфікації фахівців. Ключовими компонентами такої системи також є посилений захист критичної інфраструктури та забезпечення обміну

інформацією про загрози в реальному часі, що дозволяє своєчасно реагувати на кіберінциденти та мінімізувати їхні наслідки.

Попередні дослідження зосереджувались на психологічних профілях зловмисників та механізмах виникнення інцидентів, однак сучасні кіберзагрози вимагають розробки адаптивних методів реагування на їх постійну еволюцію. Серед найнебезпечніших атак 2022-2025 років виділяються таргетований фішинг з використанням ШІ, багаторівневі DDoS-атаки, комплексні SQL-ін'єкції, програми-вимагачі з подвійним шифруванням, атаки на ланцюги постачання та експлуатація вразливостей нульового дня[2,3]. Ці загрози потребують спеціалізованих підходів до розслідування та реагування, що враховують особливості кожного типу атак та дозволяють ефективно протидіяти сучасним кіберзагрозам.

Ефективна кіберзахисна стратегія вимагає не лише якісних методів передачі запитів до служби IT-безпеки, але й комплексних систем управління інцидентами. Організації можуть використовувати різноманітні захищені канали комунікації, від телефонних ліній з обов'язковою верифікацією до спеціалізованих порталів самообслуговування з багатофакторною автентифікацією. Ці канали доповнюються централізованими системами управління, такими як Security Operation Center (SOC) та Incident Response Platform (IRP), що забезпечують цілодобовий моніторинг, оперативний аналіз загроз та автоматизацію процесів розслідування інцидентів)[3].

Після впровадження відповідних систем реалізується чітко структурований процес обробки запитів безпеки, що охоплює чотири послідовні етапи: реєстрацію, аналіз, реагування та закриття. На етапі реєстрації черговий спеціаліст або автоматизована система створює заявку та визначає її пріоритет. Потім аналітики безпеки проводять оцінку загрози та визначають напрямок розслідування, після чого команда реагування здійснює нейтралізацію загрози з використанням спеціалізованих інструментів. Завершальний етап передбачає аналіз першопричин інциденту та оновлення бази знань, що забезпечує постійне вдосконалення системи кіберзахисту організації.

У рамках модернізованого підходу до кіберзахисту розроблено удосконалений алгоритм розслідування інцидентів (Рис.1).

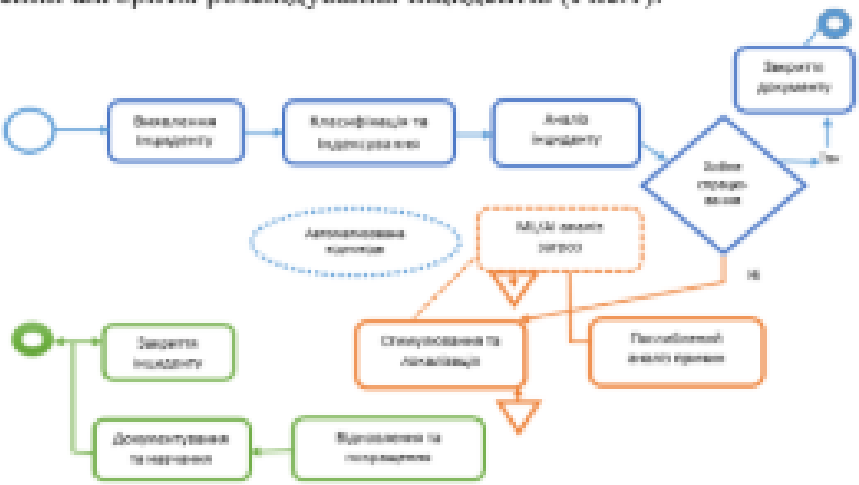

Рис.1. Алгоритм розслідування інцидентів [розроблено автором на основі 3]

План реагування на інциденти забезпечує структурований підхід до кіберзагроз через етапи виявлення, аналізу та відновлення, з чітким розподілом ролей та відповідальності команди, що мінімізує простої та фінансові втрати. Удосконалена схема на 2025-2030 роки впроваджує ML/AI аналіз загроз та автоматизовану відповідь, створюючи інтегрований підхід для сучасного захисту організацій, з перспективою розробки адаптивних механізмів реагування на новітні кіберзагрози.

1. Embroker. (2023). 2023 Must-Know Cyber Attack Statistics and Trends. https://www.embroker.com/blog/cyber-attack-statistics/
2. Globe Newswire. (2022). Cybercrime to Cost the World $10.5 Trillion Annually by 2025. https://www.globenewswire.com/news-release
3. Ceccato, V., & Newton, A. (2025). System thinking for sustainable crime prevention. Routledge. https://doi.org/10.4324/9781003281306.

Методологія криптографічного захисту інформації на основі цілочисельної, модифікованої досконалої та поліноміальної систем залишкових класів

УДК 621.395.7 (043.2)

Ігор Якименко
Західноукраїнський національний університет, jiz@wunu.edu.ua, iyakymenko@ukr.net

Представлена методологія криптографічного захисту інформаційних потоків на основі використання симетричних та асиметричних криптосистем в цілочисельній, модифікованій досконалій та поліноміальній СЗК [1-3] (складається з восьми етапів: 1) процес формування множини блоків відкритого тексту (N, $N(x)$) для цілочисельних і поліноміальних криптографічних систем; 2) Встановлення вимог щодо основних параметрів цілочисельних та поліноміальних симетричних та асиметричних криптографічних систем та захищеності інформації; 3) вибір запропонованих цілочисельних та поліноміальних криптосистем; 4) створення множини базових операцій; 5) формування набору методів виконання операцій; 6) вибір цілочисельної та поліноміальної форм СЗК; 7) програмна реалізація цілочисельних та поліноміальних симетричних та асиметричних криптосистем. Детальний опис запропонованих етапів та взаємозв'язок між ними представлено нижче.

Етап 1. Процес формування множини блоків відкритого тексту (N, $N(x)$) для цілочисельних і поліноміальних криптографічних систем в СЗК. На початковому етапі користувач повинен подати у цифровому вигляді набори блоків відкритого тексту для цілочисельного шифрування N=Ui=1M1Ni=N1,N2,...,NM1, і у вигляді поліномів для поліноміального N(x)=Ui=1M2Ni(x)=N1x,N2(x),...,NM1(x) (M1, M2 – кількість блоків відкритого тексту). У основних асиметричних криптографічних системах, таких як RSA, Рабіна та Ель-Гамаль , значення відкритого тексту не повинні перевищувати відповідний параметр відкритого ключа [4].




ЗАХІДНОУКРАЇНСЬКИЙНАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КІБЕРБЕЗПЕКИ
ГРОМАДСЬКА ОРГАНІАЦІЯ «КІБЕРБЕЗПЕКА І АВТОМАТИЗАЦІЯ»

Матеріали
науково-практичного симпозіуму

"ЗАХИСТ ІНФОРМАЦІЇ'2025"

28 листопада 2025
Тернопіль

Збірник матеріалів науково-практичного симпозіуму «Захист інформації'2025», Тернопіль, 2025. – 118с.

Адреса редакції:
Громадська організація «Кібербезпека і автоматизація»
м. Тернопіль
Контактний телефон: (066)043-42-10
e-mail: conferencekb@gmail.com

УДК 004.056.5:004.738.5:336.74

Роман ЩИПАНСЬКИЙ, Бабала ЛЮДМИЛА

Західноукраїнський національний університет

ТЕОРЕТИЧНІ ОСНОВИ БЕЗПЕКИ БЛОКЧЕЙН–ТЕХНОЛОГІЙ ТА КРИПТОВАЛЮТНИХ ТРАНЗАКЦІЙ

Вступ. Блокчейн–технологія стала фундаментальною основою сучасної цифрової економіки, забезпечуючи децентралізоване зберігання даних та проведення криптовалютних транзакцій. Однак, незважаючи на високий рівень криптографічного захисту, блокчейн–системи залишаються вразливими до різноманітних кіберзагроз, що потребує комплексного дослідження методів забезпечення їх безпеки.

Мета дослідження – проаналізувати теоретичні основи безпеки блокчейн–технологій, виявити основні вразливості та систематизувати сучасні методи захисту криптовалютних транзакцій.

1. Фундаментальні принципи блокчейн–технологій

Блокчейн являє собою розподілену базу даних, що складається з послідовно пов'язаних блоків, кожен з яких містить хеш попереднього блоку, забезпечуючи незмінність даних. Криптографічну основу становлять хеш–функції SHA–256 (Bitcoin) та Keccak–256 (Ethereum), які гарантують цілісність інформації [1]. Механізми консенсусу відіграють ключову роль у забезпеченні узгодженості транзакцій. Proof of Work (PoW) забезпечує високу безпеку через розв'язання складних обчислювальних задач, але потребує значних енергетичних витрат. Proof of Stake (PoS) зменшує енергоспоживання на основі принципу підтвердження частки володіння, однак створює ризики централізації [2].

Архітектурні особливості та масштабування. Структура блокчейну включає повні вузли (full nodes), що зберігають увесь ланцюг блоків, та легкі вузли (light nodes) для економії ресурсів (рисунок 1).

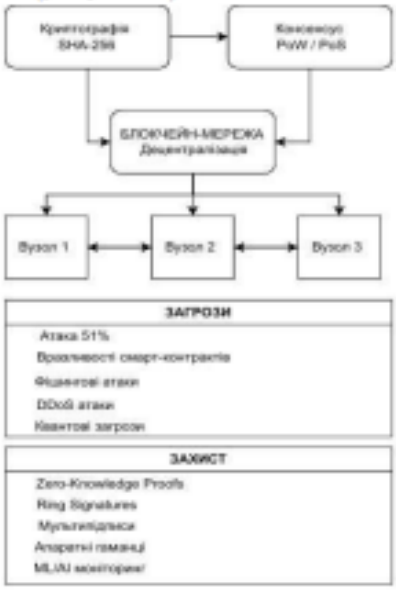

Рисунок 1 – Архітектура безпеки блокчейн–систем

Проблема масштабованості вирішується через впровадження шардингу – розділення мережі на менші частини для паралельної обробки транзакцій, та Layer 2 рішення (Lightning Network, zk–Rollups) [3].

Серед основних загроз виділяються: атака 51% (контроль над більшістю обчислювальної потужності мережі), вразливості смарт–контрактів (як у випадку інциденту з DAO, де було втрачено понад 50 млн доларів), фішингові атаки на приватні ключі користувачів, DDoS–атаки для перевантаження мережі, та потенційна загроза з боку квантових обчислень для існуючих криптографічних алгоритмів [4].

Сучасні методи захисту криптовалютних транзакцій. Забезпечення безпеки досягається через комплексний підхід [4]:

– криптографічні методи: асиметричне шифрування, цифрові підписи (ECDSA, EdDSA), Zero–Knowledge Proofs для підтвердження достовірності без розкриття інформації;

– методи анонімізації: CoinJoin, Ring Signatures, Stealth Addresses, протоколи конфіденційності (MimbleWimble);

– організаційні заходи: мультипідписи, апаратні гаманці, механізми розподіленого управління ключами (Shamir's Secret Sharing, Social Recovery Wallets). Розроблено удосконалений алгоритм розслідування кіберінцидентів з інтеграцією ML/AI аналізу загроз, що включає чотири послідовні етапи: реєстрацію з автоматизованим визначенням пріоритету, аналіз загрози з використанням машинного навчання, реагування через спеціалізовані інструменти нейтралізації, та закриття з аналізом першопричин і оновленням бази знань.

Висновки. Дослідження показало, що безпека блокчейн–систем вимагає багаторівневого підходу, що поєднує технічні (удосконалення протоколів, стійкість смарт–контрактів, криптографічні методи) та організаційні заходи (навчання користувачів, стандарти безпеки, механізми швидкого реагування). Впровадження ML/AI технологій у процеси моніторингу та реагування на інциденти створює адаптивну систему безпеки, здатну еволюціонувати паралельно з розвитком загроз. Виявлені тенденції вказують на необхідність подальшого вдосконалення методів захисту, особливо в контексті розвитку квантових обчислень та штучного інтелекту.

Перелік використаних джерел

1. Nakamoto S. Bitcoin: A Peer–to–Peer Electronic Cash System. 2008. [Ектронний ресурс].– Режим доступу: https://bitcoin.org/bitcoin.pdf

2. Zheng Z., Xie S., Dai H., Chen X., Wang H. Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services. 2018. Vol. 14. No. 4. P. 352–375.

3. Poon J., Dryja T. The Bitcoin Lightning Network: Scalable Off–Chain Instant Payments. 2016. [Ектронний ресурс].– Режим доступу: https://lightning.network/lightning–network–paper.pdf

4. Conti M., Kumar E. S., Lal C., Ruj S. A Survey of Security Threats in Blockchain Systems. ACM Computing Surveys. 2022. Vol. 55. No. 3. P. 1–36.