НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова
праця на правах рукопису

ТІХОНОВ СЕРГІЙ ВІКТОРОВИЧ

УДК 004.056.53: 004.722.45

ДИСЕРТАЦІЯ

МЕТОД КРИПТО-СТІЙКОГО ДИНАМІЧНОГО КОДУВАННЯ І РОЗПОДІЛУ ДАНИХ МОБІЛЬНОЇ AD-HOC МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ

Спеціальність 123 – Комп’ютерна інженерія
Галузь знань 12 – Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____  ____ С.В. Тіхонов

Науковий керівник:
Ситніков Валерій Степанович, доктор технічних наук, професор

Одеса – 2026

АНОТАЦІЯ

Тіхонов С.В. Метод крипто-стійкого динамічного кодування і розподілу даних мобільної ad-hoc мережі Інтернету речей. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 123 – Комп'ютерна інженерія. – Національний університет «Одеська політехніка», Міністерство освіти і науки України, Одеса, 2026.

Дисертаційна робота присвячена вирішенню актуальної науково-прикладної задачі створення проактивних методів криптостійкого динамічного кодування і розподілу даних для підвищення захисту інформації в системах та мережах Інтернету речей.

У **вступі** наведено обґрунтування актуальності теми дослідження, підкреслено недоліки наявних моделей і методів у сфері кібербезпеки, що робить їх неефективними проти невідомих раніше атак і залишає прогалини в захисті. Показано зв’язок роботи з науковими програмами, планами і темами. Визначено об’єкт, предмет, мету, задачі і методи дослідження. Наведено наукову новизну та практичне значення отриманих результатів. Висвітлено особистий внесок здобувача в наукових публікаціях за темою дисертаційних досліджень. Наведені дані про апробацію, використання і впровадження результатів дисертаційної роботи, про публікації здобувача на тему дисертації, а також наведена інформація про структуру і обсяг дисертації.

У **першому розділі** дисертаційної роботи проведено аналітичний огляд проблеми захисту інформації і управління розподілом даних в комп’ютерних системах та мережах у зв’язку із стрімким впровадженням мобільних мереж Інтернету речей (IoT) у промисловість, оборону і повсякденне життя. На основі системного підходу, досліджено розвиток еталонних моделей Інтернету у контексті інформаційної безпеки та сучасні криптографічні технології Інтернету речей.

Проаналізовано існуючі протоколи і стандарти захисту в безпровідних інтерфейсах WiFi, а також моделі і методи управління цифровими потоками в мобільних ad-hoc мережах Інтернету речей.

Аналіз робіт у сфері захисту інформації і управління розподілом даних в комп'ютерних системах та мережах показав, що на поточний момент створено ефективні еталонні безпекові моделі Інтернету речей від організацій IoT-WF, NIST. Досягнуто високої криптографічної стійкості шифрів і крипто-протоколів (шифр AES, протокол GCMP, автентифікація SAE, стандарт WPA3). Існують швидкі моделі та алгоритми оптимізації потоків даних в мережах ("Push-Relabel"). Потужним інструментом захисту і оптимізації мереж IoT є використання ШІ. Разом з тим, існують невирішені питання та виклики, зокрема, консерватизм застарілих технологій, що стримує втілення нових безпекових рішень поверх наявних, а також реактивний підхід до захисту даних в комп'ютерних мережах (виявлення загроз по факту). Окрім того, є недостатньою швидкодія оптимізації потоків реального часу в мобільних ad-hoc мережах. Залишається можливість додаткового збільшення максимального потоку шляхом динамічної ре-конфігурації каналів зв'язку при швидких змінах структури мережі.

Зроблено висновок, що існуючі рішення недостатні для комплексної підтримки сучасних вимог до кодування, захисту і управління даними в мобільних ad-hoc мережах IoT. Є потреба у пост-криптографічному захисті даних незалежно від існуючої апаратної підтримки, проактивних методах динамічного кодування захисту і розподілу даних, усунення вразливості застарілих операційних технологій, а також зменшення впливу людського фактору у питаннях кібербезпеки.

У зв'язку з цим, поставлено завдання щодо створення методу крипто-стійкого динамічного кодування на основі багаторівневої формальної граматики і побудови недетермінованої прикладної машини Тьюринга, а також комп'ютерних алгоритмів крипто-стійкого динамічного кодування і розподілу даних ad-hoc мережі Інтернету речей.

У **другому розділі** дисертаційної роботи обґрунтовано основні теоретичні положення щодо синтезу моделі і методу крипто-стійкого динамічного кодування даних для підвищення рівня кіберзахисту мобільної ad-hoc мережі Інтернету речей. Сформульовано перші три наукові результати роботи.

Досліджено еволюцію екосистеми робототехніки “Robotics Eco-System”, а також методи і моделі автоматизованого проектування роботизованих мобільних комплексів і платформ. Проаналізовано особливості кодування і структуризації даних у технологіях локальних мереж Ethernet та її бездротового аналогу технології WiFi. Обґрунтовано багаторівнева формальна граматика для гнучкої структуризації фреймів мобільної ad-hoc мережі Інтернету речей.

Розглянуто узагальнену функціональну схему дистанційного управління мобільною платформою Інтернету речей. Розроблено трирівневу структуру інтерфейсів роботизованого мобільного комплексу, а також формальну граматику побудови фреймів в каналі зв’язку з трійковим лінійними сигналами для нестандартного утворення цифрового потоку з окремих незалежних функціональних елементів у довільному порядку і складі. На основі цього, удосконалено модель динамічного кодування даних мобільної ad-hoc мережі Інтернету речей, що дозволяє збільшити ефективність каналу у 1.5÷6.6 разів та ускладнити його злам за рахунок скорочення службової інформації і непередбачуваного формату даних (**перший науковий результат**).

Визначено систему команд і узагальнену формальну граматику для прикладної машини Тьюринга з правилами семантики потокового кодування фреймів. Розроблено недетерміновану машину Тьюринга для структуризації даних в каналі зв’язку, *відмінну тим*, що через випадкові інтервали часу, кодові кортежі команд перемішуються і змінюють свою семантику у порядку відомому лише конкретному адресату. Таким чином, *отримала подальший розвиток* методика застосування теорії автоматів, *що дозволяє* створювати різноманіття алгоритмів криптографічного захисту інформації незалежно від існуючих апаратно-програмних засобів (**другий науковий результат**).

Вперше запропоновано спосіб динамічної синхронізації профільного ключа крипто-стійкого кодування між бортовим контролером мобільного об'єкту та базовою станцією дистанційного управління на основі недетермінованої машини Тьюринга, *відмінний тим*, що на кожну місію мобільного об'єкту генерується і закладається в пам'ять обох сторін унікальна випадкова послідовність профільних ключів, а номер ключа у цій послідовності відіграє роль відкритого ключа для передачі по каналу зв'язку, *що дозволяє* підвищити проактивний кіберзахист мережі, спростити його апаратно-програмну реалізацію і підвищити ефективність каналів зв'язку, а також обмежити людський фактор у питаннях конфіденційності (**третій науковий результат**).

У **третьому розділі** дисертаційної роботи розроблено комп'ютерний алгоритм для крипто-стійкого динамічного кодування в мобільній ad-hoc мережі Інтернету речей (IoT) на основі недетермінованої машини Тьюринга, теоретичні засади якого сформульовані у розділі 2. Побудовано блок-схему алгоритму і програмний модуль TMSend на мові Python. Наведено типові приклади симуляції недетермінованої машини Тьюринга NdTM у середовищі Python/Mac OS. Визначено три базові профіля роботи роботи алгоритму:

а) «гнучка структура» (без крипто-захисту): гнучке формування структури фрейму при відсутності кіберзагроз для зменшення службової інформації та підвищення інформаційної ефективності каналу зв'язку; дає виграш інформаційної ефективності у 1.5÷6.6 рази.

б) «крипто-захист»: динамічне крипто-стійке кодування фрейму змінним ключом що синхронізується між взаємодіючими сторонами; забезпечує еквівалентний до існуючих методів («нормальний») рівень захисту простими засобами незалежно (або на додаток) до традиційних методів шифрування; забезпечує достатню для ad-hoc мережі стійкість до зламу і виграш інформаційної ефективності у 2.5÷9.6 рази (в залежності від розміру даних і типу інкапсуляції).

в) «Пост-шифрування»: додаткове крипто-кодування корисних та службових даних тимчасовим ключем, номер якого передається відкритим параметром і визначає значення ключа в секретному масиві, що наперед випадково згенеровано і завантажено в пам’ять взаємодіючих сторін; даний механізм реалізує підвищений проактивний рівень захисту даних на додаток до відомих методів шифрування і дозволяє підвищити рівень захисту відомих методів; підвищує рівень захисту незалежно і на додаток до існуючих методів.

Досліджено структурно-параметричні характеристики нормалізованого планарного графу ad-hoc мережі (ST-графу). Систематизовано і розширено базові формули комбінаторики з шістьма новими функціями для обчислення параметрів ST-графу за кількістю його вершин.

Розглянуто структуру шляхів ST-графу з точки зору побудови дерева пошуку найкоротших шляхів.

Удосконалено методи рішення комбінаторних задач на графах *шляхом* обґрунтування рекурсивної побудови дерева пошуку найкоротших шляхів на вільно-орієнтованому графі мережі та комбінаторної формули від числа вершин графу для визначення необхідної кількості ітерацій, *що дозволяє* прискорити алгоритми оптимізації потоків даних в мережах, а також залучити штучний інтелект до реалізації таких алгоритмів (**четвертий науковий результат**).

Розроблено комбінаторну модель оптимізації розподілу даних ad-hoc мережі Інтернету речей з використанням штучного інтелекту, а також комп’ютерний алгоритм динамічного розподілу даних в мережі IoT.

Імплементація цього алгоритму розширює можливості адаптації ad-hoc мережі Інтернету речей до динамічно змінюваних факторів для *максимально ефективного використання наявних ресурсів* (енерго-запаси мобільних об’єктів, канали зв’язку, мережеві інтерфейси тощо).

Це підсилює функціональну стійкість мережі, і зокрема, рівень кіберзахисту, що важливо при організації спеціальних місій для групи автономних мобільних роботизованих платформ, які об’єднуються в ad-hoc мережу змінної топології і метрики.

У висновках зазначено основні результати дисертаційного дослідження.

Сукупність наукових результатів дисертаційного дослідження дозволяє розв'язати актуальну науково-прикладну задачу, що полягає у створенні проактивних методів криптостікого динамічного кодування і розподілу даних для підвищення захисту інформації в системах та мережах Інтернету речей.

Основним науковим результатом дисертаційної роботи є створення методу крипто-стійкого динамічного кодування і розподілу даних мобільної ad-hoc мережі Інтернету речей для підвищення захисту інформації. *На відміну від існуючих*, даний метод застосовує недетерміновану машину Тьюринга зі змінною у часі системою команд для гнучкого формування структури даних в каналі зв'язку, а також комбінаторну модель оптимального розподілу даних для рекурсивної побудови дерева пошуку найкоротших шляхів на вільно-орієнтованому планарному графі з використанням штучного інтелекту, *що дозволяє* підвищити проактивний кіберзахист мережі, спростити його апаратно-програмну реалізацію, підвищити ефективність каналів зв'язку, обмежити людський фактор у питаннях конфіденційності, збільшити швидкодію адаптивної реконфігурації розподілу даних та підсилити функціональну стійкість в умовах динамічно змінних факторів впливу.

У межах основного наукового результату роботи отримано наступне.

1) *Удосконалено* модель динамічного кодування даних мобільної ad-hoc мережі Інтернету речей *шляхом* розробки і застосування трирівневої структури інтерфейсів роботизованого мобільного комплексу та відповідної формальної граматики побудови фреймів в каналі зв'язку з трійковим лінійними сигналами для нестандартного утворення цифрового потоку з окремих незалежних функціональних елементів у довільному порядку і складі, *що дозволяє* збільшити ефективність каналу у 1.5÷6.6 разів та ускладнити його злам за рахунок скорочення службової інформації і непередбачуваного формату даних.

2) *Отримала подальший розвиток* методика застосування теорії автоматів шляхом розробки недетермінованої машини Тьюринга для структуризації даних в каналі зв'язку, *відмінна тим*, що через випадкові інтервали часу, кодові кортежі команд перемішуються і змінюють свою семантику у порядку відомому лише конкретному адресату, *що дозволяє* створювати різноманіття алгоритмів криптозахисту незалежно від існуючих апаратно-програмних засобів.

3) *Вперше запропоновано* спосіб динамічної синхронізації профільного ключа крипто-стійкого кодування між бортовим контролером мобільного об’єкту та базовою станцією дистанційного управління на основі недетермінованої машини Тьюринга, *відмінний тим*, що на кожну місію мобільного об’єкту генерується і закладається в пам’ять обох сторін унікальна випадкова послідовність профільних ключів, а номер ключа у цій послідовності відіграє роль відкритого ключа для передачі по каналу зв’язку, *що дозволяє* підвищити проактивний кіберзахист мережі, спростити його апаратно-програмну реалізацію і підвищити у 2.5÷9.6 разів ефективність каналів зв’язку, а також обмежити людський фактор у питаннях конфіденційності.

4) *Удосконалено* методи рішення комбінаторних задач на графах *шляхом* обґрунтування рекурсивної побудови дерева пошуку найкоротших шляхів на вільно-орієнтованому графі мережі та відповідної комбінаторної формули від числа вершин графу для визначення необхідної кількості ітерацій, *що дозволяє* прискорити у 1.3÷1.5 рази алгоритми оптимізації потоків даних в мережах, а також залучити штучний інтелект до реалізації таких алгоритмів.

Результати дисертації сприяють підвищенню рівня кіберзахисту систем та мереж Інтернету речей в умовах зростання кіберзагроз та складності атак. Впровадження результатів дисертації дозволить підвищити стійкість та надійність функціонування критично важливих систем та мереж, що є необхідною передумовою сталого розвитку цифрового суспільства.

Практично значущими є наступні результати роботи.

– Для суб’єктів критичної інфраструктури та промислових підприємств, метод крипто-стійкого динамічного кодування та розподілу даних може бути використано для надійного стійкого захисту від атак типу DoS, ін’єкцій шкідливого коду та несанкціонованого доступу. Це дозволить запобігти збою енергосистем типу Smart Grid, промислових IoT-мереж і транспортних систем.

– Для розробників та виробників IoT-пристроїв, алгоритми та протоколи, що використовують штучний інтелект для адаптивного керування кодуванням, можуть бути інтегровані у програмне забезпечення нових пристроїв.

Це забезпечить проактивний захист і можливість автономно протистояти новим, раніше невідомим загрозам без необхідності втручання оператора.

– Для розробників та користувачів мобільних роботизованих об`єктів та ad-hoc мережам. Запропонована методологія забезпечує підвищену стійкість зв'язку та маскування трафіку в умовах активної радіоелектронної боротьби. Динамічний розподіл даних та адаптивне кодування дозволяють протидіяти навмисним перешкодам, запобігати втраті управління та інформації в роях дронів та інших автономних системах, що працюють у ворожому середовищі. Це є критично важливим для безпеки та ефективного виконання бойових, розвідувальних та рятувальних місій.

– Для наукових установ та закладів вищої освіти, розроблена методологічна база слугує теоретичним підґрунтям для подальших наукових досліджень у сфері кібербезпеки децентралізованих систем. Результати роботи можуть бути використані для модернізації навчальних курсів та спеціалізованих дисциплін з інформаційної безпеки та штучного інтелекту.

Окремі результати роботи впроваджено на ТОВ «Радуга-N» для послуг «Розумний дім»» (спосіб динамічної синхронізації поточного профільного ключа між контролером МО і БС, Додаток Б), у навчальні дисципліни ДУІТЗ: «Інформаційні технології кодування і забезпечення завадостійкості систем» (методи теорії автоматів і недетермінованої машини Тьюринга для крипто-стійкого кодування), «Інтернет речей» (контекстно-визначена модель взаємодії відкритих систем для дослідження проблем кібербезпеки IoT), «Методи та системи штучного інтелекту» (алгоритм комбінаторної оптимізації розподілу даних з використанням штучного інтелект (Додаток В), а також у навчальний процес та наукову діяльність Національного університету «Одеська політехніка» (Додатки Г, Д).

Основні результати дисертаційної роботи представлені і обговорені на 7 міжнародних науково-практичних конференціях (НПК) та відображені у 11 наукових публікаціях, у тому числі, 4 статті у наукових фахових виданнях що індексуються у НМБД Scopus (у співавторстві), 7 опублікованих тез доповідей на міжнародних НПК (з них 5 одноосібних).

Дисертаційна робота підготовлена у відповідності до законів України «Про основні засади забезпечення кібербезпеки України» від 27.03.2025, «Про захист інформації в інформаційно-комунікаційних системах» від 20.04.2025, «Про Стратегію кібербезпеки України» (рішення Ради національної безпеки та оборони України від 14.05.2021). Тема роботи корелюється з пунктом 1.2.9.5 («Дослідження та розроблення методів інформаційної безпеки комп'ютерних систем і мереж») положень Національної академії наук України «Основні наукові напрями та найважливіші проблеми фундаментальних досліджень у галузі природничих, технічних і гуманітарних наук Національної академії наук України на 2024–2028 роки» No8 від 10.01.2024.

Тема дисератційної роботи відповідає науковим планам і тематиці досліджень, а також програмам підготовки студентів Національного університету «Одеська політехніка» у сфері кібербезпеки, інтелектуальних систем та робототехніки. Дисертація є складовою частиною НДР No 199-64 «Моделі, методи розробки та апаратні рішення високочастотних компонентів програмованих мобільних систем» Інституту штучного інтелекту та робототехніки Національного університету «Одеська політехніка» (No держреєстрації 0121U110245).

Ключові слова: комп'ютерні системи та мережі, Інтернет речей, мобільна ad-hoc мережа, кодування, захист інформації, інтерфейс зв'язку, дистанційне управління, робототехніка, формальна граматика, машина Тьюринга, динамічний розподіл даних, нормалізований граф, комбінаторика, штучний інтелект.

Список публікацій здобувача за темою дисертації

1. Тіхонов С.В. Захищене кодування мережевих інтерфейсів у розподілених системах інтернету речей на основі багаторівневих формальних граматик. *Proc. of the VII International Scientific and Practical Conference «Science and Technology: Problems, Prospects and Innovations»* (Osaka, 13-15 April, 2023). С. 174-180. URL: https://sci-conf.com.ua/wp-content/uploads/2023/04/SCIENCE-AND-TECHNOLOGY-PROBLEMS-PROSPECTS-AND-INNOVATIONS-13-15.04.23.pdf.
2. Тіхонов С. Ієрархічний підхід до кодування даних на основі адаптивної формальної граматики. *Proc. of the XIV International Scientific and Practical Conference "Science and Practice: Implementation to Modern Society"* (Manchester, 26-28 April, 2023), № 152. P. 559-563. URL: https://archive.interconf.center/plugins/generic/pdfJsViewer/pdf.js/web/viewer.html.2C1040%2C899%5D.
3. Ситніков В.С., Тіхонов С.В. Принципи захисту інформації у комп'ютерних системах Інтернету речей на основі ієрархічного кодування у формальних граматиках. *Збірник тез 24-ї Міжнародної НПК «Сучасні інформаційні та електронні технології» CIET/MIET-2023* (Одеса, 29-31 травня 2023). С. 31-32. URL: https://old.tkea.com.ua/siet/archive/2023/31.pdf.
4. Тіхонов С.В. Формальна граматика машини Тьюринга для завадостійкого кодування фреймів Ethernet на основі трійкових лінійних сигналів. *Proc. of XII Internetional Scientific and Practical Conference «Innovations and Prospects in Modern Science»* (Stockholm, 20-22 November, 2023). С. 315-325. URL: https://sci-conf.com.ua/wp-content/uploads/2023/11/INNOVATIONS-AND-PROSPECTS-IN-MODERN-SCIENCE-20-22.11.23.pdf.
5. Тіхонов С. Питання кібербезпеки в базових технологіях Інтернету речей. *Proc. of XII International Scientific and Practical Conference "Current Challenges of Science and Education"* (Berlin, 29-31 July, 2024). С. 165-171.

URL: https://sci-conf.com.ua/wp-content/uploads/2024/07/CURRENT-CHALLENGES-OF-SCIENCE-AND-EDUCATION-29-31.07.24.pdf.

6. Тіхонов В., Тіхонов С., Яворська О. Дослідження комбінаторних потокових задач на планарному графі комп'ютерної мережі. *Proc. of XI International Scientific and Practical Conf. "Science in the Modern World: Innovations and Challenges"* (Toronto, 10-12 July 2025). С. 45-52. URL: https://sci-conf.com.ua/wp-content/uploads/2025/07/SCIENCE-IN-THE-MODERN-WORLD-INNOVATIONS-AND-CHALLENGES-12-14.06.2025.pdf.
7. Тіхонов С.В. Підвищення стійкості ad-hoc мережі IoT на основі швидкого алгоритму розподілу даних з використанням ШІ. *Proc. of V International Scientific and Practical Conference "International Experience in Scientific Rresearch"* (Chicago, 18-20 December 2025). P. 300-308. URL: https://sci-conf.com.ua/wp-content/uploads/2025/12/INTERNATIONAL-EXPERIENCE-IN-SCIENTIFIC-RESEARCH-18-20.12.25.pdf.
8. Тіхонов В., Тахер А., Тіхонов С., Шулакова К., Глущенко В., Чайка А. Розробка машини Тюрінга для високозахищеного кодування даних на канальному рівні Інтернету речей. *Hochschule Anhalt Edition "Applied Innovations in Information and Communication Technology" (ICAIIT)*. 2024. V. 12, № 1. P. 1-10. ISSN 2199-8876. DOI: http://dx.doi.org/10.25673/115635.
9. Тіхонов В., Сіменс Е., Васіліу Є., Ситніков В., Тахер А., Тихонова О., Шулакова К., Тіхонов С. Контекстно-визначена модель взаємодії відкритих систем для дослідження проблем кібербезпеки Інтернету речей. *Hochschule Anhalt Edition "Proceedings of the International Conference on Applied Innovations in IT" (ICAIIT).* 2024. V. 12, № 2, P. 35-44. ISSN 2199-8876. DOI: http://dx.doi.org/10.25673/118112.
10. Тіхонов В., Тахер А., Шулакова К., Тіхонов С., Демченко О. Розробка машини Тюрінга для високозахищеного дистанційного керування мобільною платформою IoT. *Springer Nature Link, Lecture Notes in Networks and Systems, ICAIIT- 2024*. 2025. V. 1338. P. 61–80. ISSN 2367-3370, 2367-3389 (electronic). DOI: https://doi.org/10.1007/978-3-031-89296-7_5.

11. Тіхонов В., Ситніков В., Тіхонов С. Удосконалений алгоритм оптимізації потоків даних на нормалізованому біполярному планарному вільноорієнтованому мережевому графі з використанням штучного інтелекту. *Hochschule Anhalt Edition "Proceedings of the International Conference on Applied Innovations in IT" (ICAIIT)*. 2025. V. 13, № 5. P. 7–14. ISSN 2199-8876. DOI: http://dx.doi.org/10.25673/122800.

ABSTRACT

Tikhonov S.V. Method of crypto-resistant dynamic coding and data distribution in mobile ad-hoc network of the Internet of Things. – Qualification scientific paper as a manuscript.

Thesis for the PhD degree in specialty 123 – Computer engineering. – Odesa Polytechnic National University, Ministry of Education and Science of Ukraine, Odesa, 2026.

The dissertation is dedicated to solving the actual scientific and applied task of creating proactive methods of crypto-resistant dynamic data encoding and distribution to increase information protection in the Internet of Things systems and networks.

The **introduction** justifies the relevance of the research topic, and emphasizes the shortcomings of existing models and methods in the cybersecurity area, which makes them ineffective against previously unknown attacks and leaves gaps in protection. The study aligns with institutional research programs, plans, and projects. The object, subject, aim, objectives, and methods of the research are defined. The scientific novelty and practical significance of the results are presented. The section highlights the author's personal contribution to the co-authored scientific publications on the research topic. The applicant personal contribution is highlighted in the scientific publications on the dissertation research. The data of dissertation results presentation, application, and implementation is given along with the applicant's publications on dissertation topic and the info of the dissertation structure and scope.

In the **first chapter**, an analytical review was conducted regarding the issue of information protection and data distribution control in computer systems and networks amid the rapid implementation of mobile Internet of Things (IoT) networks into industry, defence, and everyday life. Development of the IoT reference models was investigated by the systems approach in the context of information security and modern cryptographic technologies of the Internet of Things. Existing protocols and security standards in WiFi wireless interfaces were analysed, as well as models and methods for digital flows control in mobile ad-hoc Internet of Things networks.

An analysis of the information protection and data distribution in computer systems and networks realm studies revealed that currently effective reference security models for the Internet of Things have been developed by the IoT-WF and NIST. High cryptography resistance of ciphers and crypto protocols has been achieved (AES cipher, GCMP protocol, SAE authentication, WPA3 standard). There are fast models and algorithms for data flows optimizing in networks ("Push-Relabel").

A powerful tool for protecting and optimizing IoT networks is the use of AI. However, there are unresolved issues and challenges, in particular, the conservatism of legacy technologies, which hinders the implementation of new security solutions on the top of existing ones, as well as a reactive approach to data protection in computer networks (detection threats de-facto). In addition, the speed of real-time flows optimization in mobile ad-hoc networks is insufficient. The possibility remains of further increasing the maximum flow by dynamically reconfiguring communication channels during rapid changes in the network structure.

It is concluded that existing solutions are insufficient to comprehensive support the modern requirements for encryption, protection and data management in mobile ad-hoc IoT networks. There is a need for post-cryptographic data protection regardless of existing hardware support, proactive methods of dynamic data encryption, protection and distribution, remediation of vulnerabilities in legacy operational technologies, as well as reducing the human factor impact in cybersecurity issues.

Considering this, the goal was set to develop a method of crypto-stable dynamic coding based on a multi-level formal grammar and the construction of a non-deterministic applied Turing machine, as well as computer algorithms for crypto-resistant dynamic coding and data distribution in an ad-hoc Internet of things network.

In the **second chapter,** the main theoretical propositions are substantiated regarding the synthesis of a model and method of crypto-resistant dynamic data encoding to increase the cyber-security of a mobile ad-hoc Internet of things network. The first three scientific results of the work are formulated. The evolution of the "Robotics Eco-System" is studied, as well as methods and models of automated design the robotic mobile complexes and platforms.

The features are analysed of data coding and structuring in Ethernet local area network technologies and its wireless analogue WiFi technology. A multi-level formal grammar is substantiated for flexible frame structuring in a mobile ad-hoc IoT network. A generalized functional scheme is considered of remote control the Internet of Things' mobile platform. A three-level interfaces structure is developed for a robotic mobile complex, as well as a formal grammar for constructing frames in a communication channel with ternary linear signals for non-standard formation of a digital stream from by separate independent functional elements in an arbitrary order and composition. Based on this, the model of dynamic data coding is improved for a mobile ad-hoc Internet of Things network, that allows increasing the channel efficiency by 1.5÷6.6 times and making it more difficult to hack due to the reduction of overhead information and unpredictable data format (**first scientific result**).

A command system and a generalized formal grammar have been defined for an applied Turing machine with the frame stream coding semantics rules. A nondeterministic Turing machine has been developed for data structuring in a communication channel, which differs in that at random intervals of time, the code tuples of commands are permutated and change their semantics in an order known only to particular addressee. Thus, the methodology of applying automata theory *was further developed*, which *allows* creating a variety of algorithms for cryptographic info protection regardless of existing hardware and software (**second scientific result**).

Firstly a method of dynamic synchronization of the crypto-resistant coding profile key between the on-board controller of a mobile object and the remote control base station with nondeterministic Turing machine is proposed, distinguished in that for each mission of a mobile object, a unique random sequence of profile keys is generated and stored in the memory of both parties, and the key number in this sequence plays the role of a public key for transmission over the communication channel, which allows increase proactive cyber protection of the network, simplify its hardware and software implementation and increase the efficiency of communication channels, as well as mitigate the human factor in privacy issues (**third scientific result**).

In the **third** chapter, a computer algorithm for crypto-stable dynamic coding in a mobile ad-hoc Internet of Things network is developed based on a nondeterministic Turing machine, the theoretical foundations of which formulated in section 2. A flowchart of the algorithm and a TMSend program Python module are constructed. Typical examples of simulation of a nondeterministic Turing machine NdTM in the Python/Mac OS environment given. The following three basic profiles of the algorithm's operation defined. a) "Flexible structure" (without crypto-protection): flexible formation of the frame structure in the absence of cyber threats to reduce overhead data and increase efficiency of the communication channel; provides a gain in information efficiency of 1.5÷6.6 times. b) "Crypto-protection": dynamic crypto-stable encoding of a frame with a variable key that is synchronized between the interacting parties; provides an equivalent level of protection to existing methods ("normal") by simple means independently (or in addition) to traditional encryption methods; provides sufficient resistance to hacking for an ad-hoc network and a gain in information efficiency of 2.5÷9.6 times (depending on the data size and type of encapsulation). c) "Post-encryption": additional crypto-encoding of useful/service data with a temporary key, the number of which is transmitted by a public parameter and determines the value of the key in a secret array, which is randomly generated in advance and loaded into the memory of the interacting parties; this mechanism implements an increased proactive level of data protection in addition to known encryption methods and allows to increase the level of protection of known methods; increases protection independently and in addition to existing methods.

The structural and parametric characteristics of the normalized planar graph of an ad-hoc network (ST-graph) are investigated. The basic combinatorics formulas are systematized and expanded with six new functions for calculating the parameters of the ST-graph by the number of its vertices. The structure of the paths of the ST-graph is considered in the context of constructing a shortest path searching tree.

The methods of combinatorial tasks on graphs solving *have been improved due to* justification the recursive construction of a shortest path searching tree on a free-oriented network graph along with the combinatorial formula for the graph vertices

number to determine the required number of iterations, which allows acceleration the algorithms for data flows optimizing in networks, as well as involve artificial intelligence into implementation of such algorithms (**fourth scientific result**).

A combinatorial model for optimizing data distribution in an ad-hoc Internet of Things network using artificial intelligence has been developed, as well as a computer algorithm for dynamic data distribution in an IoT network. The implementation of this algorithm expands the possibilities of adapting an ad-hoc Internet of things network to dynamically changing factors forthe most efficient use of available resources(energy reserves of mobile objects, communication channels, network interfaces, etc.). This enhances the functional stability of the network, and in particular, the cyber protection level, which is important when organizing special missions for a group of autonomous mobile robotic platforms that are combined into an ad-hoc network of variable topology and metrics.

In the conclusions, the main results of the dissertation research are indicated. The set of scientific results of the dissertation research allows solving a relevant scientific and applied task of creation proactive methods of cryptographic dynamic coding and data distribution to increase information protection in Internet of things systems and networks.

The *main scientific result* of the dissertation is creation a method of crypto-resistant dynamic data encoding and distribution in a mobile ad-hoc Internet of things network to increase information protection. *Unlike existing*, this method uses a nondeterministic Turing machine with a time-varying command system for flexible data structure formation in the communication channel, as well as a combinatorial model of optimal data distribution for recursive construction a shortest path searching tree on a free-oriented planar graph using artificial intelligence, *which allows* increase the proactive cyber defense of the network, simplify its hardware and software implementation, increase the efficiency of communication channels, mitigate the human factor in privacy issues, increase the speed of adaptive reconfiguration of data distribution, and enhance functional stability under dynamically changing impacts.

The following was obtained within the main scientific result of the work.

1) *Improved* dynamic data encoding model for mobile ad-hoc Internet of things network *due to* development and application of a three-level interface structure of a robotic mobile complex and the corresponding formal grammar for constructing frames in a communication channel with ternary linear signals for non-standard formation of a digital stream from separate independent functional elements in an arbitrary order and composition, which allows increase the channel efficiency by 1.5÷6.6 times and make it more difficult to crack due to reduction the overhead and unpredictable data format.

2) *Further advanced* a methodology of automata theory applying *by developing* a nondeterministic Turing machine for structuring data in a communication channel, *different in that* at random intervals of time, the code tuples of commands are permitted and changing their semantics in an order known only to the particular addressee, which allows to create a variety of algorithms for cryptographic information protection regardless of existing hardware and software.

3) *Firstly proposed* a method of dynamic synchronization of a crypto-stable encoding profile key between an on-board controller of a mobile object based on a non-deterministic Turing machine and a remote control base station, *different in that* for each mission of a mobile object, a unique random sequence of profile keys is generated and stored in the memory of both parties, and the key number in this sequence plays the role of a public key for transmission over the communication channel, *which allows* increase proactive cyber defense of the network, simplify its hardware and software implementation and increase the efficiency of communication channels by 2.5÷9.6 times, as well as limit the human factor in matters of confidentiality.

4) *Improved* methods of combinatorial tasks on graphs solving *due to* justification the recursive construction of a shortest path searching tree on a free-oriented network graph along with the combinatorial formula for the graph vertices number to determine the required number of iterations, *which allows* acceleration the algorithms for data flows optimizing in networks, as well as involve artificial intelligence into implementation of such algorithms.

The results of the dissertation facilitatei increasing the cyber protection of Internet of things systems and networks in the face of growing cyber threats and attacks complexity. Implementation of dissertation results enables resilience and functional reliability strengthen of critical systems and networks, which is a necessary prerequisite for sustainable development of a digital society.

The following results of the work are practically significant.

– For critical infrastructure entities and industrial enterprises, the method of crypto-resistant dynamic data encoding and distribution can be used for reliable and resilient protection against DoS attacks, malicious code injections, and unauthorized access. This will prevent the failure of Smart Grid energy systems, industrial IoT networks, and transportation systems.

– For IoT device developers and manufacturers, the algorithms and protocols that use artificial intelligence for adaptive coding control can be integrated into the software of new devices. This will provide proactive protection, allowing systems to autonomously confront unknown threats without the need for operator intervention.

– For developers and users of mobile robotic objects and ad-hoc networks. The proposed methodology enables increased communication stability and traffic masking in conditions of active electronic warfare. Dynamic data distribution and adaptive coding allow to counteract intentional interference, prevent the loss of control and information in swarms of drones and other autonomous systems operating in a hostile environment. This is critically important for the safety and effective performance of combat, reconnaissance and rescue missions.

– For scientific institutions and higher education institutions, the developed methodological base serves as a theoretical basis for further scientific research in the field of cybersecurity of decentralized systems. The results of the work can be used to modernize training courses and specialized disciplines in information security and artificial intelligence.

Some results of the work were implemented at Raduga-N Ltd for the Smart Home services (a method of dynamic synchronization of the current profile key between the MO controller and the BS, Appendix B), in the academic disciplines of

the SUITT: “Information technologies of coding and ensuring system protection” (methods of the theory of automata and nondeterministic Turing machine for crypto-resistant coding), “Internet of Things” (a context-defined interaction model of open systems for studying IoT cybersecurity problems), “Artificial intelligence methods and systems” (algorithm for combinatorial optimization of data distribution using artificial intelligence (Appendix B), as well as in the educational process and scientific activities of the Odesa Polytechnic National University (Appendixes G, D).

The main results of the dissertation work were presented and discussed at 7 international scientific and practical conferences (SPC) and reflected in 11 scientific publications, including 4 articles in scientific professional publications indexed in Scopus (in co-authorship), 7 published abstracts of reports at international SPCs (5 of them solo).

The dissertation work was prepared in accordance with the laws of Ukraine “On the Basic Principles of Ensuring Cybersecurity of Ukraine”, 03.27.2025, “On Information Protection in Information and Communication Systems” dated 04/20/2025, “On the Cybersecurity Strategy of Ukraine” (decision of the National Security and Defence Council of Ukraine, 05/14/2021). The topic of the work correlates with paragraph 1.2.9.5 (“Research and development of methods for information security of computer systems and networks”) of the provisions of the National Academy of Sciences of Ukraine “Main scientific directions and important problems of fundamental research in the field of natural, technical and humanities of the National Academy of Sciences of Ukraine for 2024-2028” No. 8, 01.10.2024.

The dissertation topic corresponds to the scientific plans and research topics, as well as the training programs of students of the National University "Odesa Polytechnic" in the field of cybersecurity, intelligent systems and robotics. The dissertation is a component of the Research and Development Project No. 199-64 "Models, development methods and hardware solutions of high-frequency components of programmable mobile systems" of the Institute of AI and Robotics of the National University "Odesa Polytechnic" (state registration № 0121U110245).

Keywords: computer systems and networks, Internet of things, mobile ad-hoc network, coding, information security, communication interface, remote control, robotics, formal grammar, Turing machine, dynamic data distribution, normalized graph, combinatorics, artificial intelligence.

List of publications of the applicant on the topic of the dissertation

1. Tikhonov S.V. Secure coding of network interfaces in distributed Internet of Things systems based on multi-level formal grammars. *Proc. of the VII International Scientific and Practical Conference «Science and Technology: Problems, Prospects and Innovations»* (Osaka, 13-15 April, 2023). P. 174-180. URL: https://sci-conf.com.ua/wp-content/uploads/2023/04/SCIENCE-AND-TECHNOLOGY-PROBLEMS-PROSPECTS-AND-INNOVATIONS-13-15.04.23.pdf.
2. Tikhonov Serhii. Hierarchical data coding approach based on adaptive formal grammar. *Proc. of the XIV International Scientific and Practical Conference "Science and Practice: Implementation to Modern Society"* (Manchester, 26-28 April, 2023), № 152. P. 559-563. URL: https://archive.interconf.center/plugins/generic/pdfJsViewer/pdf.js/web/viewer.html.2C1040%2C899%5D.
3. Sitnikov V.S., Tikhonov S.V. Principles of information protection in Internet of Things computer systems based on hierarchical coding in formal grammars *Zbirnyk tez 24-th Msznarodnoi NPK «Suchasni informaciyni ta electronni technologii» CIET/MIET-2023* (Odesa, 29-31 May 2023). P. 31-32. URL: https://old.tkea.com.ua/siet/archive/2023/31.pdf.
4. Tikhonov S.V. Formal Turing machine grammar for resistant coding of Ethernet frames based on ternary linear signals. *Proc. of XII Internetional Scientific and Practical Conference «Innovations and Prospects in Modern Science»* (Stockholm, 20-22 November, 2023). C. 315-325. URL: https://sci-conf.com.ua/wp-content/uploads/2023/11/INNOVATIONS-AND-PROSPECTS-IN-MODERN-SCIENCE-20-22.11.23.pdf.

5. Tikhonov S. The cybersecurity issues in basic Internet of things technologies. *Proc. of XII International Scientific and Practical Conference "Current Challenges of Science and Education"* (Berlin, 29-31 July, 2024). P. 165-171. URL: https://sci-conf.com.ua/wp-content/uploads/2024/07/CURRENT-CHALLENGES-OF-SCIENCE-AND-EDUCATION-29-31.07.24.pdf.
6. Tikhonov V., Tikhonov S., Yavorska O. Research of combinatorial flow problems on a planar graph of a computer network. *Proc. of XI International Scientific and Practical Conf. "Science in the Modern World: Innovations and Challenges"* (Toronto, 10-12 July 2025). C. 45-52. URL: https://sci-conf.com.ua/wp-content/uploads/2025/07/SCIENCE-IN-THE-MODERN-WORLD-INNOVATIONS-AND-CHALLENGES-12-14.06.2025.pdf.
7. Tikhonov S.V. Increasing the resilience of an ad-hoc IoT network based on a fast data distribution AI driven algorithm. *Proc. of V International Scientific and Practical Conference "International Experience in Scientific Rresearch"* (Chicago, 18-20 December 2025). P. 300-308. URL: https://sci-conf.com.ua/wp-content/uploads/2025/12/INTERNATIONAL-EXPERIENCE-IN-SCIENTIFIC-RESEARCH-18-20.12.25.pdf.
8. Tikhonov V., Taher A., Tikhonov S., Shulakova K., Hluschenko V., Chaika A. Turing Machine Development for High-secure Data Link Encoding in the Internet of Things Channel. *Hochschule Anhalt Edition "Applied Innovations in Information and Communication Technology" (ICAIIT)*. 2024. V. 12, № 1. P. 1-10. ISSN 2199-8876. DOI: http://dx.doi.org/10.25673/115635.
9. Tikhonov V., Siemens E., Vasiliu Y., Sitnikov V., Taher A., Tykhonova O., Shulakova K., Tikhonov S. Context-Defined Model of Open Systems Interaction for IoT Cybersecurity Issues Study. *Hochschule Anhalt Edition "Proceedings of the International Conference on Applied Innovations in IT" (ICAIIT).* 2024. V. 12, № 2, P. 35–44. ISSN 2199-8876. DOI: http://dx.doi.org/10.25673/118112.
10. Tikhonov V., Taher A., Shulakova K., Tikhonov S., Demchenko O. Turing Machine Development for High Protected Remote Control of the IoT Mobile Platform. *Springer Nature Link, Lecture Notes in Networks and Systems, ICAIIT-*

2024. 2025. V. 1338. P. 61–80. ISSN 2367-3370, 2367-3389 (electronic). DOI: https://doi.org/10.1007/978-3-031-89296-7_5.

11. Tikhonov V., Sitnikov V., Tikhonov S. An Improved AI-Driven Algorithm of Data Flow Optimization on the Normalized Bipolar Planar Free-Oriented Network Graph. *Hochschule Anhalt Edition "Proceedings of the International Conference on Applied Innovations in IT" (ICAIIT).* 2025. V. 13, № 5. P. 7–14. ISSN 2199-8876. DOI: http://dx.doi.org/10.25673/122800.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Скорочення	Значення	Примітка
БПЛА		Безпілотний літальний апарат.
МТ	Машина Тьюринга	Turing Machine.
НСД		Не санкціонований доступ. Найбільший спільний дільник
ПЛІС	FPGA	Програмована логічна інтегральна схема.
СКК	Сигнально-кодові конструкції	Спосіб кодування інформації на фізичному рівні OSI.
ТСК	Таймерні сигнальні конструкції	Тип кодування інформації на фізичному рівні OSI.
ШІ	Штучний Інтелект	Artificial Intelligence.
ШНМ	Штучна нейронна мережа	Artificial Nyron Network (ANN).
AAS	As a Service	Маркетинговий продукт як сервісна послуга.
AD	Address vector	Адресний вектор параметрів фрейму (адреси призначення та відправника).
AEMFP	Almost Equal Maximum Flow Problem	«Майже рівний максимальний потік» – розширена задача оптимізації потоків електроенергії.
AES	Advanced Encryption Standard	Блочний 128-біт алгоритм симетрич. шифрування з ключами 128/192/256 біт, діючий станд. США (2000), с.16.
AI	Artificial Intelligence	Штучний інтелект (ШІ)
AIV	Autonomous Intelligent Vehicles	Автономні інтелектуальні транспортні засоби.
AMT	Applied Turing Machine	Прикладна ашини Тьюринга.
AP	Access Point	Точка доступу (WiFi).
API	Application Programming Interface	Інтерфейс прикладного програмування.
APK	Android Package Kit	Архів програм під Android/iOS “Ionic Framework” SDK на мовах HTML, CSS, JavaScript.
ARP	Autonomous Robotics Platrorm	Автономна роботизована платформа.
ARP	Address Resolution Protocol	Протокол рівня L2 OSI для пошуку IP-адреси хоста по його MAC-адресі в локальній мережі Ethernet.

ASCII	American Standard Code -Information Interchange	Американський стандартний код для обміну інформацією, 128 символів (+ розширення до 256 символів).
AUV	Autonomous Underwater Vehicles	Автономний підводний транспортний засіб (без оператора управління).
BD	Backdoors	“Чорні двері”, вірусний код, що скасовує нормальні процедури автентифікації доступу.
BCS	Basic Control System	Базова система управління.
BER	Bit Error Rate	Рівень (ймовірність) бітових помилок.
BFS	Breadth First Search	Метод пошуку шляхів вщир, с. 38.
Brownfield OT	Brownfield Operational Technologies	Проблема застарілих операційних технологій («Освоєна і кинута земля»).
BSS	Basic Service Set	Базовий набір послуг ad-hoc мережі WiFi (приєднання/ вихід/зміна), с. 61
CA	Certification Authority	Центр сертифікації клієнтів мережі доступу.
CAN	Controller Area Network	Стандарт повідомлень електроніки автомобіля та вбудованих систем промислової автоматизації.
CBC	Cipher Block Chaining	Режим шифрування методом AES-256.
CCMP	Counter cipher message protocol	Counter mode with Cipher block chaining Message authentication code Protocol.
CEK	Current Encryption Key	Поточний ключ крипто-кодування цифрового потоку фрейма машиною Тьюринга NdTM.
CEN	Comité Européen de Normalisation	Європе́йський комітéт зі стандартиза́ції.
cGAN	Conditional Generative Adversarial Net	Умовна генеративно-змагальна мережа.
CIA	Confidentiality, Integrity, Availability	Конфіденційність, цілісність, доступність (концепція захисту інформації).
Code_tab	Coding Table	Кодова таблиця розміром 2×15, де строка 0 – теги команд MT, строка 1 – теги підстановки (кортеж Kt).
CPU	Central Processor Unit	Центральний процесорний пристрій.

CRC	Cyclic Redundancy Check	Циклічна контрольна сума (фрейму).
CSA	Connectivity Standards Alliance	Спільнота розробників стандартів IoT (раніше Zigbee Alliance)
CSS	Cascading Style Sheets	Каскадні таблиці стилів документа на мовіах HTML, XML, SVG, MathML, XHTML для відображення на екрані, папері, мовленні тощо.
DAP	Data Access Point	Область пам'яті процесора з адресом доступу.
DARPA	Defense Advanced Research Projects Agency	Дослідницьке агентство з новітніх військових технологій Міністерства оборони США.
DES	Data Encryption Standard	Блочний 64-біт алгоритм симетричн. шифрування з ключом 56 біт - старий стандарт США (1970).
DIX	DEC, Intel, Xerox	Спільна розробка компаній DEC, Intel і Xerox - Ethernet-2 (станд. DIXv2.0, 1982).
DL	Deep Learning	Метод глибокого навчання ШНМ
DLL	Data Link Layer	Канальний рівень моделі OSI.
DMAC	Destination MAC address	MAC адреса призначення (отримувача фрейму).
DMM	Dynamic Migration Management	Проактивна концепція використання AI для «активної оборони» («Динамічна міграція даних»).
DSA	Digital Signature Algorithm	Алгоритм цифрового підпису.
DTC	Digital Twin Consortium	Міжнародний консорціум з Індустріального Інтернету речей.
DTMF	Dual Tone Multi-Frequency	Двох-тональний багато-частотний декодер.
DWPG	Directed Weighted Planar Graph	Орієнтований зважений планарний граф.
EAP-TLS	Extensible Authentication Protocol-Transport Layer Security	Розширюваний протокол автентифікації захисту транспортного рівня.
ECC	Elliptic Curve Cryptography	Алгоритм асиметричного шифрування з дискретним логарифмом на еліптичних кривих, для х мобільних систем.
ECDH	Elliptic-curve Diffie–Hellman	Протокол узгодження ключів через незахищений канал.

ECDSA	Elliptic Curve Digital Signature Algorithm	Алгоритм цифрового підпису з відкритим ключем (PKC) на еліптичних кривих.
ECU	Electronic Control Unit	Електронні блоки контроля в автомобілях.
EMFP	Equal Maximum Flow Problem	«Рівний максимальний потік» - задача оптимізація потоків енергії.
ET	Ether Type	Поле «Тип фрейму» Ethernet.
eUtility	External Utility	Зовнішня утиліта (службова програма)
F-Bus	FieldBus	«Польова шина» (1980) - технологія IoT в концепції “Industry 4.0”.
FC (FCS)	Frame Check (Sum)	Контрольна сума фрейму.
FDC	Frame Dynamic Code	Динамічний код фрейму машини Тьюринга NdTM.
FFA	Ford–Fulkerson Algorithm	Алгоритм Форда-Фалкерсона для максимального потоку планарного орієнтованого ST-графу.
FG	Formal Grammar	Формальна граматика.
FIPS 197	Federal Information Processing Standard	Федеральний стандарт обробки інформації.
FL	Federated Learning	Федеративне навчання AI на розподілених наборах даних.
FPC	Frame Parameter Code	Параметричний коду фрейму.
FPGA	Field Programmable Gate Array	Програмована логічна інтегральна схема (ПЛІС).
FSA	Finite State Automaton	Скінченний автомат.
FSC	Frame Streaming Code	Потоковий код фрейму.
FSM	Finite State Machine	Машина з кінечним числом своїх станів.
FWG	Free-oriented Weighted Graph	Вільно-орієнтований зважений граф.
GCMP	Galois/Counter Mode Protocol	Потоковий режим симетричного блокового шифру, який водночас забезпечує конфіденційність інформації та автентифікацію її джерела (напр., AES-128-GCM).
GPS	Global Positionong System	Глобальна навігаційна система.
GPT	Generative Pre-trained Transformer	Генеративний попередньо-тренований перетворювач.

GPU	Graphical Processor Unit	Графічний процесорний пристрій, с. 115.
GRE	Generic Routing Encapsulation	Протокол маршрутної інкапсуляції IP-пакетів у пакети-обгортки для з'єднання типу «точка-точка», с. 69
GUI	Graphical User Interface	Графічний інтерфейс користувача.
HEX	Hexadecimal value for the character	Шістнадцяткове значення символу.
HTML	Hypertext Markup Language	Мова гіпертекстової розмітки тегами (командами) розмітки.
HTTP	Hyper Text Transfer Protocol	Основний протокол прикладного рівня у світовій павутині World Wide Web (WWW).
ICV	Integrity Check Value	Значення перевірки цілісності.
IDE	Integrated Development Environment	Інтегроване середовище автоматизованого проектування.
IEEE	Institute of Electrical and Electronics Engineers	Провідна всесвітня організація з розвитку і стандартизації в комп'ютерних системах та мережах
IETF	Internet Engineering Task Force	Міжнародна організація «Цільова група з розвитку Інтернету».
IFG	Inter Frame Gap	Захисний міжкадровий інтервал.
IIC	Industry IoT Consortium	Консорціум Інтернету рчей (нині у складі Digital Twin Consortium).
IISF	Документ IIC	Industrial Internet Security Framework.
IIoT	Industrial IoT	Індустріальний Інтернет речей.
IIRA	Industrial Internet Reference Srchitecture	Еталонна архітектура промислового Інтернету..
IMT	Inverse Math Task	Зворотна математична задача.
IMU	Inertial Measurement Unit	Інерційний вимірювальний блок.
INL	Intelligent Node Labelling	Інтелектуальний спосіб маркування вузлів для пошуку максимального потоку на орієнтованому графі.
InOut_tab	Input-Output Table	Таблиця (2×N) вхідних та вихідних даних; де строка 0 – вхідний пакет In_pack(N), строка 1 – вихідний пакет Out_pack(N).
InPack	Input Packet	Пакет вхідних даних як одномірний масив символів.

IoTWF	IoT World Forum	Світова спільнота розробників Інтернету речей.
IPM	Interior Point Method	Метод внутрішніх точок для оптимізації потоків в мережах.
ISM	Industrial, Scientific, Medical	Радіо частоти 900/868 МГц та 2.4 Ггц.
ISO	International Standardization Organization	Міжнародна організація по стандартизації.
IST	Input Sequential Text	Вхідний текстовий пакет прикладного рівня (дані + службові команди) для крипто-кодування машиною Тьюринга NdTM.
ITU	International Telecommunication Unit	Міжнародний союз електрозв'язку.
IV	Initiation Vector	Вектор ініціації, с. 32.
Kt_index	Key index	Індекс (номер) кортежу поточного системного ключа k_t в масиві SEK з K кортежів.
LC	Line Code	Крипто-кодований машиною NdTM вхідний текстовий пакет IST (дані + службові команди) - послідовність вихідних трійкових лінійних сигналів канального рівня.
LLC	Logical Link Control	Другий підрівень канального рівня OSI (логічне керування каналом): синхронізація, контроль помилок.
LLN	Logical Link Number	Логічний номер з'єднання.
LoRaWAN	Long Range Wide Area Network	Протокол бездротового надійного і недорогого зв'язку в IoT.
LPWAN	Low-Power Wide-Area Network	Енергоефективна мережа великого радіуса дії для повільної передачі даних.
LTE	Long Term Evolution	Стандарт мобільного зв'язку 4G.
M-Bus	Meter BUS	Технологія Інтернету речей для дистанційного зчитування лічильників побутових послуг (1991), с. 21.
MAC	Media Access Control	Підрівень канального рівня OSI - Контроль доступу до середовища.

MC	Mobile Controller	Контролер мобільної платформи MP для взаємодії зі станцією RCS дистанційного управління.
MFP	Max Flow Problem	Задача про максимальний потік мережі.
MIC	Message Integrity Check	Перевірка цілісності повідомлення.
Mii	Mii -codes	Неформальна назва бінарних лінійних кодів (від ігрових Mii-гравців).
MIT	Massachusetts Institute of Technology	Массачусетський технологічний інститут США.
ML	Machine Learning	Машинне навчання.
MMO	Mealy Machine with Ouput	Скінченний автомат типу «Машина Мілі з виходом» (переробляє вхідні символи у вихідні).
MoveIT	Robotics Web-platform	Відкрита веб-платформа для розробки робото-технічних додатків.
MP	Mobile Platform	Мобільна платформа.
MPLS	Multi Protocol lable Switching	Багато-протокольна комутація по мітках.
MPRC	Mobile Platform Remote Control	Система дистанційного управління мобільною платформою.
MPS	Mobile Platform Simulator	Симулятор мобільної платформи.
MQTT	Message Queuing Telemetry Transport	Протокол взаємодії в реальному часі з низькою затримкою в системах IoT з обмеженими ресурсами.
MRP	Mobile Robotics Platform	Мобільна роботизована платформа.
MSK	Mission System Key	Системний ключ місії (одноразовий для кожної місії) набір віпадкових ключей крипто-кодування.
NAV	Network Allocation Vector	Вектор розподілу мережі для обмеження часу доступу до середовища.
NdTM	Nondeterministic Turing Machine	Недетермінована машина Тьюринга.
NdTM-DDE	Nondeterministic Turing Machine for	Недетермінована машина Тьюринга для динамічного крипто-кодування даних (Dynamic Data Encryption).
NGN	Next Generations Networks	Мережі наступних поколінь.

NIST	National Institute of Standards and Technology	Національний інститут стандартів і технологій США.
NIST SP	NIST Special Publication	Спеціальний випуск Національного інституту стандартів і технологій США.
NRZ	Non-Return to Zero	«Без повернення до нуля» - бінарні лінійні коди без обов'язкового перемикання сигналу в окремих символах.
O(f)	O(function of arguments)	Функціія обчислювальної складності алгоритму (кількість операцій або час виконання).
ORB-SLAM	Oriented fast and Rotated Brief SLAM	Метод покращення нечіткого зображення під водою.
OSI	Open System Interconnection	Взаємодія відкритих систем (7-рівнева еталонна модель Інтернету від ISO).
OST	Output Sequential Text	Декодований машиною NdTM пакет IST (без службових команд), отриманий через вхідні лінійні трійкові сигнали.
P-Bus	ProfiBus	«Польова шина» (1987).
PAM2	Pulse Amplitude Modulation 2	Імпульсна амаплітудна модуляція з двома рівнями амплітуди лінійного сигналу.
PDA	Personal Digital Assistant	Персональний цифровий асистент.
PHP	Hypertext Pre-processor	Раніше "Personal Home Page". Мова програмування веб-додатків і сайтів за допомогою протоколу HTML.
PKC	Public Key Cryptography	Криптографія з відкритим ключем.
PL	Payload	Поле корисних даних фрейму.
PM	Post Machine	Машина Поста (з двома символами алфавіту).
PR	Peamble	Преамбула в структурі фрейму FDC машини Тьюринга NdTM.
PRB	Preamble	Преамбула синхронізації фрейму Ethernet (12 байт «1010101.....01**0**».
PSK	Pre-Shared Key	Попередньо узгоджений спільний ключ шифрування.

QNX OS	Unix-like real-time operating system	Unix-подібна операційна система реального часу для вбудованих систем Інтернету речей.
QoS	Quality of Service	Якість обслуговуваання.
QUIC	Quick UDP Internet Connections	Просунутий протокол транспортного рівня як альтернатива до TCP (гнучкість, вбудована безпека, висока продуктивність), с. 23.
RC	Remote Controller	Контролер станції дистанційного управління RCS для взаємодії з мобільною платформою MP, с. 85
RC4	Rivest Cipher 4	Застарілий потоковий шифр із симетричним ключем (Ronald Rivest, 1987).
RCS	Remote Control Station	Станція дистанційного управління.
RES	Robotics Eco-System	Екосистема робототехніки (апаратно-програмні засоби і стандарти).
RFC	Request for Comments	Сімейство стандартів Міжнародної організації з розвитку Інтернету IETF.
RFR	Random Forest Regression	Алгоритм машинного навчання з учителем через ансамбль незалежних дерев рішень.
RIF	Robotics Interoperability Model	Модель взаємодії об’єктів у роботизованому комплексі.
ROS	Robotics Real-time Operating System	Операційна система реального часу.
ROV	A remotely operated vehicle	Дистанційно керований транспортний засіб (умбілікус, кабель-пуповина).
RSA	Rivest–Shamir–Adleman	Асиметричний шифр (1970) на принципі факторизації великих чисел (розкладання їх на прості множники).
RTD	Real-time Telemetry Data	Телеметричні дані реального часу.
RTS	Request To Send	«Запит на відправлення» для зменшення колізій кадрів WiFi.
RZ	Return to Zero	Тип бінарних лінійних кодів.
SAE	Simultaneous Authentication of Equals	Спосіб автентифікації (станд. IEEE 802.11s-2011/2012), стійкий до атак “offline dictionary”.
SDK	Software Development Kit	Набір засобів для розробки програмних продуктів.

SDN	Software-Defined Networking	Програмно-визначені мережі з API контролерами для контролю потоків мережі через базову інфраструктуру.
SEK	System Encryption Key	Системний ключ крипто-кодування (набір випадкових перестановок команд машини Тьюринга NdTM).
SF	Start Frame	Делімітер логічної синхронізації цифрового потоку в структурі фрейму FDC машини Тьюринга NdTM.
SFD	Start Frame Delimiter	Стартовий байт «10101011» між преамбулою синхронізації та першим байтом тіла фрейму Ethernet.
SHAP	Shapley Additive Explanations	Адитивні пояснення Shapley.
SIMD	Single Instruction Multiple Data	Метод паралельних (векторних) обчислень (напр., в графічних процесорах).
SKA	Shared Key Authentication	Атентифікація спільним ключом (паролем доступу) у крипто-протоколі WEP (застарілий).
SL/UL/RL	Supervised/ /Unsupervised/ Reinforcement/ Learning	Методи машинного навчання з учителем/без учителя/з підкріпленням.
SLAM	Simultaneous Localization and Mapping	Одночасна локалізація та картографування.
SMAC	Source MAC address	MAC адреса джерела (відправника фрейму).
SNAP	Sub-Network Access Protocol	Сімейство стандартів технології Ethernet IEEE-802.3x.
SPN	Substitution-Permutation Network	Структура шифрування: «Мережа заміщення-перестановки».
SSH	Secure Shell	Протокол безпечної передачі даних через слабо захищений канал «робот-смартфон» (напр., радіо-інтерфейс WiFi, взаємодія через Інтернет.
SSIM	Structural Similarity Index Measure	Міра (індекс) структурної подібності для оцінки візуальної якості зображень.
SSL	Secure Sockets Layer	Старий крипто-протокол (1990); сучасний - TLS).

STAP	ST-Alternative Paths	Альтенативні ST-шляхи між полюсами S і T на графі
STP/FWG	ST-Planar Free-oriented Weighted Graph	ST-планарний вільно=-орієнтований зважений граф
TCP/IP	Protocol suite	Стек протоколів сучасного Інтернету, в якому є протоколи IP та TCP.
TEB	Timed Elastic Band	Оптимізатор локального шляху для автономної багато-точкової навігації у відомому середовищі.
TEK	Temporary Encryption Key	Тимчасовий ключ крипто-кодування блоків корисних даних і службових параметрів машини Тьюринга NdTM.
TKIP	Temporal Key Integrity Protocol	Протокол шифрування з тимчасовим ключом (перехідний варіант між WEP та WPA.
TLS	Transport Layer Security	Сучасний крипто-протокол (1999 - TLS 1.0).
TM	Turing Machine	Машина Тьюринга.
TMcom	Turing Machine Commands	Система команд машини Тьюринга як двомірний масив символів розміром (2×15).
TMF	Turing Machine Frame	Фрейм машини Тьюринга.
TMsend	Turing Machine Send	Програмний модуль для крипто-кодування і передачі в лінію зв'язку віхдних пакетів корисних даних і службових команд (IST).
ToDS/FromDS	To/From Destination	Біти ToDS/FromDS «до місця призначення» та «від місця призначення».
TTL	Transistor–Transistor Logic	Сімейство логічних схем на біполярних транзисторах.
UDP	User Datagram Protocol	Протокол транспортного рівня стеку TCP/IP.
UGV	Unmanned Ground Vehicle	Безпілотний наземний транспортний засіб.
USB	Universal Serial Bus	Універсальна послідовна шина (інтерфейс).
VMPC	Variable Modified Permutation Composition	Змінна модифікована композиція перестановок (спосіб шифрування).
VP	Virtual Pilot	Віртуальний пілот мобільної платформи.

VRG	Virtual Reality Glasses	Окуляри віртуальної реальності.
VRP	Vehicle Routing Problem	Маршрутизації транспортних засобів.
VTOL	Vertical Take-Off and Landing	Вертикальний зльот/посадка з горизонтально орієнтованим фюзеляжам.
WAN	Wide Area Network	Мережа великого масштабу
WEP	Wired Equivalent Privacy	Застарілий протокол крипто-захисту мереж (WiFi 802.11).
WiFi	Wireless Fidelity	«Бездротова якість» - технологія безпровідного зв’язку.
WPA	WiFi Protected Access	Протокол крипто-захисту мереж WiFi.
WPAN		Персональна мережа з топологією ad-hoc mesh.
XAI	Explainable Artificial Intelligence	«Зрозумілий» штучний інтелект що використовує адитивні пояснення Shapley (SHAP).
ZigBee	Технологія Інтернету речей	Автоматичний збір даних з пристроїв IoT (10÷100 м на радіо частотах ISM.

ВСТУП

Актуальність теми

В умовах стрімкого поширення новітніх інформаційних технологій (ІТ) у повсякденне життя, промисловість та оборонну, актуальним стає проблема захисту персональних даних і кібербезпеки інфраструктурних об’єктів і систем. Конфіденційність стала серйозним викликом для сучасного Інтернету речей (IoT), оскільки хакерські цифрові системи можуть впроваджуватися у реальні фізичні об'єкти і наносити значну шкоду. Кіберзахист IoT ускладнюється великим обсягом даних і документів різного типу, кожен з яких є потенційним джерелом загроз (віруси, фішинг, скомпрометовані мережі WiFi, файли зображень, шахрайські USB-накопичувачі). Ланками вразливості IoT є радіоканали, програмне забезпечення контролерів, пристрої комутації і розподілу трафіку, файли «exe» та «com» з вірусними закладками (bookmarks) що містять посилання на хакерські веб-ресурси, тощо.

Незважаючи на значні успіхи вітчизняних та зарубіжних науковців, існуючі підходи у сфері кібербезпеки мають певні недоліки, такі як реактивний характер, фрагментарність, ігнорування людського фактору та інші, що робить їх неефективними проти нових невідомих раніше атак, ускладнює менеджмент, залишає прогалини в захисті і вимагає надмірних ресурсів для інтеграції та підтримки. При цьому традиційні підходи переважно зосереджені на технічних рішеннях, але недостатньо уваги приділяють захисту від соціальної інженерії, фішингу та інших атак, що використовують людську необізнаність чи помилки. Багато успішних кібератак починається саме з компрометації користувача. Принципово нову загрозу безпеці IoT представляє розвиток штучного інтелекту, але водночас, він надає додаткові можливості протидії цим викликам.

Внаслідок сказаного вище, виникає *актуальна науково-прикладна задача* – створення проактивних методів криптостійкого динамічного кодування і розподілу даних для підвищення захисту інформації в системах та мережах Інтернету речей.

Зв'язок роботи з науковими програмами, планами, темами

Дисертаційна робота підготовлена у відповідності до законів України «Про основні засади забезпечення кібербезпеки України» від 27.03.2025 [1], «Про захист інформації в інформаційно-комунікаційних системах» від 20.04.2025 [2], «Про Стратегію кібербезпеки України» (рішення Ради національної безпеки та оборони України від 14.05.2021 [3].

Тема дослідження пов’язана з науковими напрямами, сформульованими в п. 1.2.9 та 1.2.9.5 – теорія та комп’ютерні технології інформаційної безпеки «Основні наукові напрями та найважливіші проблеми фундаментальних досліджень у галузі природничих, технічних і гуманітарних наук НАН України на 2024–2028 роки», затверджених постановою Президії НАНУ від 10.01.2024 №8 [4],

а також корелюють з Постановою КМ України № 942 від 7.09.2011 [5] із змінами, внесеними постановою КМ No 463 від 9.05.2023 «Про затвердження переліку пріоритетних тематичних напрямів наукових досліджень і науково-технічних розробок на період до 2023 року» (Розділ «Інформаційні та комунікаційні технології», підрозділ «Інформаційно-комунікаційні та радіоелектронні системи та технології, засоби РЕБ для забезпечення національної безпеки і оборони. Інформаційна безпека та кібербезпека» [6].

Тема дисератційної роботи відповідає науковим планам і тематиці досліджень, а також програмам підготовки студентів Національного університету «Одеська політехніка» у сфері кібербезпеки, інтелектуальних систем та робототехніки. Дисертація є складовою частиною НДР № 199-64 «Моделі, методи розробки та апаратні рішення високочастотних компонентів програмованих мобільних систем» Інституту штучного інтелекту та робототехніки Національного університету «Одеська політехніка» (№ держреєстрації 0121U110245) [7].

Мета і завдання дослідження. *Метою* дисертаційної роботи є підвищення захисту інформації в мобільних ad-hoc мережах Інтернету речей на основі моделей і методу крипто-стійкого динамічного кодування та розподілу даних для протидії від несанкціонованого втручання в умовах змінюваної структури і обмежених ресурсів мережі. Для досягнення цієї мети в роботі поставлено і вирішено наступні завдання.

1) Огляд існуючих методів кіберзахисту і управління даними в системах та мережа Інтернету речей, обґрунтування актуальності теми дисертації.

2) Синтез методу крипто-стійкого динамічного кодування даних мобільної ad-hoc мережі Інтернету речей, у тому числі: удосконалення моделі динамічного кодування даних на основі багаторівневої формальної граматики; розробка прикладної машини Тьюринга для структуризації даних в каналі зв’язку; побудова недетермінованої машини Тьюринга для крипто-стійкого динамічного кодування даних.

3) Розробка та дослідження комп’ютерних алгоритмів крипто-стійкого динамічного кодування і розподілу даних ad-hoc мережі інтернету речей, у тому числі, розробка комбінаторної моделі для оптимізації розподілу даних ad-hoc мережі Інтернету речей з використанням штучного інтелекту. Оцінка ефективності запропонованих моделей і методу.

Об'єкт і предмет дослідження

Об’єктом дослідження дисертаційної роботи є процеси кодування і управління потоками даних у комп’ютерних системах та мережах. *Предметом* дослідження є теоретичні засади і практичні методи захищеного кодування та управління даними в мобільних ad-hoc мережах Інтернету речей (IoT).

Методи дослідження

Для розв’язання поставлених задач в роботі використано логічний аналіз, теорія інформації і кодування даних, теорія автоматів, методи захисту інформації в мережах зв'язку, теорія графів, комбінаторика, штучний інтелект. Для перевірки результатів використано методи комп’ютерного моделювання і порівняння з базовими підходами за метриками якості алгоритмів.

Наукова новизна отриманих результатів

Основним науковим результатом дисертаційної роботи є створення методу крипто-стійкого динамічного кодування і розподілу даних мобільної ad-hoc мережі Інтернету речей для підвищення захисту інформації. *На відміну від існуючих*, даний метод застосовує недетерміновану машину Тьюринга зі змінною у часі системою команд для гнучкого формування структури даних в каналі зв'язку, а також комбінаторну модель оптимального розподілу даних з використанням штучного інтелекту для рекурсивної побудови дерева пошуку найкоротших шляхів на вільно-орієнтованому планарному графі, *що дозволяє* підвищити проактивний кіберзахист мережі, спростити його апаратно-програмну реалізацію та підвищити ефективність каналів зв'язку, обмежити людський фактор у питаннях конфіденційності, збільшити швидкодію розподілу даних та підсилити функціональну стійкість в умовах динамічно змінних факторів впливу.

У межах основного наукового результату роботи отримано наступне.

1) Удосконалено модель динамічного кодування даних мобільної ad-hoc мережі Інтернету речей *шляхом* розробки і застосування трирівневої структури інтерфейсів роботизованого мобільного комплексу та відповідної формальної граматики побудови фреймів в каналі зв'язку з трійковим лінійними сигналами для нестандартного утворення цифрового потоку з окремих незалежних функціональних елементів у довільному порядку і складі, *що дозволяє* збільшити ефективність каналу та ускладнити неавторизований доступ *за рахунок* скорочення службової інформації і непередбачуваного формату даних.

2) *Отримала подальший розвиток методика* застосування теорії автоматів *шляхом* розробки недетермінованої машини Тьюринга для структуризації даних в каналі зв'язку, *відмінна* тим, що через випадкові інтервали часу, кодові кортежі команд перемішуються і змінюють свою семантику у порядку відомому лише конкретному адресату, що *дозволяє* створювати різноманіття алгоритмів криптографічного захисту інформації незалежно від існуючих апаратно-програмних засобів.

3) *Вперше запропоновано спосіб* динамічної синхронізації профільного ключа крипто-стійкого кодування між бортовим контролером мобільного об’єкту та базовою станцією дистанційного управління на основі недетермінованої машини Тьюринга, *відмінний* тим, що на кожну місію мобільного об’єкту генерується і закладається в пам’ять обох сторін унікальна випадкова послідовність профільних ключів, а номер ключа у цій послідовності відіграє роль відкритого ключа для передачі по каналу зв’язку, *що дозволяє* підвищити проактивний кіберзахист мережі, спростити його апаратно-програмну реалізацію і підвищити ефективність каналів зв’язку, а також обмежити людський фактор у питаннях конфіденційності.

4) *Удосконалено методи* рішення комбінаторних задач на графах *шляхом* обґрунтування рекурсивної побудови дерева пошуку найкоротших шляхів на вільно-орієнтованому графі мережі та відповідної комбінаторної формули від числа вершин графу для визначення необхідної кількості ітерацій, *що дозволяє* прискорити алгоритми оптимізації потоків даних в мережах, а також залучити штучний інтелект до реалізації таких алгоритмів.

Практичне значення отриманих результатів.

Результати дисертації, в цілому, мають прикладне значення для підвищення рівня кіберзахисту систем та мереж IoT в умовах зростання кіберзагроз та складності атак. Впровадження результатів дисертації дозволить підвищити стійкість та надійність функціонування критично важливих мереж, що є необхідною передумовою для безпечного розвитку цифрового суспільства.

Зокрема, практично значущими є наступні результати.

– Для суб’єктів критичної інфраструктури та промислових підприємств, метод крипто-стійкого динамічного кодування та розподілу даних може бути використано для надійного стійкого захисту від атак типу DoS, ін’єкцій шкідливого коду та несанкціонованого доступу. Це дозволить запобігти збою енергосистем типу Smart Grid, промислових IoT-мереж і транспортних систем.

– Для розробників та виробників IoT-пристроїв, алгоритми та протоколи, що використовують штучний інтелект для адаптивного керування кодуванням, можуть бути інтегровані у програмне забезпечення нових пристроїв.

Це забезпечить проактивний захист і можливість автономно протистояти новим, раніше невідомим загрозам без необхідності втручання оператора.

– Для розробників та користувачів мобільних роботизованих об'єктів та ad-hoc мереж, запропонована методологія уможливлює підвищену стійкість зв’язку та маскування трафіку в умовах активної радіоелектронної боротьби. Динамічний розподіл даних та адаптивне кодування дозволяють протидіяти навмисним перешкодам і запобігати втраті управління та інформації в роях дронів та інших автономних системах, що працюють у ворожому середовищі. Це є критично важливим для безпеки та ефективного виконання бойових, розвідувальних та рятувальних місій.

– Для наукових установ та закладів вищої освіти, розроблена методологічна база слугує теоретичним підґрунтям для подальших наукових досліджень у сфері кібербезпеки децентралізованих систем. Результати роботи можуть бути використані для модернізації навчальних курсів та спеціалізованих дисциплін з інформаційної безпеки та штучного інтелекту.

Окремі результати роботи впроваджено на ТОВ «Радуга-N» для послуг «Розумний дім»» (спосіб динамічної синхронізації поточного профільного ключа між контролером МО і БС, Додаток Б), у навчальні дисципліни ДУІТЗ: «Інформаційні технології кодування і забезпечення завадостійкості систем» (методи теорії автоматів і недетермінованої машини Тьюринга для крипто-стійкого кодування), «Інтернет речей» (контекстно-визначена модель взаємодії відкритих систем для дослідження проблем кібербезпеки IoT), «Методи та системи штучного інтелекту» (алгоритм комбінаторної оптимізації розподілу даних з використанням штучного інтелект (Додаток В), а також у навчальний процес та наукову діяльність Національного університету «Одеська політехніка» (Додатки Г, Д).

Особистий внесок здобувача. Основні наукові положення, висновки і рекомендації, що містяться в дисертації та виносяться на захист, отримано здобувачем особисто.

В роботі [3] здобувач дослідив по літературних джерелах ланки вразливості об’єктів і технологій IoT, а також питання кібербезпеки в сучасних моделях IoT. В роботі [6] здобувачу належить емпіричне обчислення дерев пошуку найкоротших шляхів на нормалізованому вільно-орієнтованому ST-планарному графу мережі для кількості вершин від 3 до 7. В роботі [8] здобувачу належить формалізація статичної і динамічної структури фрейму для кодування в термінах формальної граматики машини Тьюринга, а також принцип випадкового перемішування кодів команд машини Тьюринга. В роботі [9] здобувачу належить аналіз літературних джерел з питань кібербезпеки в технологіях і моделях Інтернету речей, а також принцип побудови контекстно-визначеної ієрархічної моделі у дослідженнях проблеми кібербезпеки. В роботі [10] здобувачем розроблено синтаксис недетермінованої машини Тьюринга, а також запропоновано принцип використання унікальної випадкової послідовності профільних ключів машини Тьюринга для взаємодіючих сторін. В роботі [11] здобувачу належить обчислення дерева пошуку шляхів і побудова навчальної структури даних для ШІ Google Gemini на нормалізованому планарному графі з кількістю вершин від 3 до 7.

Апробація результатів дисертації

Основні результати дисертаційної роботи представлені та обговорені на наступних 7 міжнародних науково-практичних конференціях.

1) VII International Scientific and Practical Conf. «SCIENCE AND TECHNOLOGY: PROBLEMS, PROSPECTS AND INNOVATIONS», Osaka, Japan, April 13-15, 2023 [8].

2) XIV International Scientific and Practical Conf. “SCIENCE AND PRACTICE: IMPLEMENTATION TO MODERN SOCIETY”, Manchester, United Kingdom, April 26-28, 2023 [9].

3) Міжнародна науково-практична конф. «СУЧАСНІ ІНФОРМАЦІЙНІ ТА ЕЛЕКТРОННІ ТЕХНОЛОГІЇ», Одеса, Україна, травень 29-31, 2023 [10].

4) XII International Scientific and Practical Conf. «INNOVATIONS AND PROSPECTS IN MODERN SCIENCE», Stockholm, November 20-22, 2023 [11].

5) XII International Scientific and Practical Conf. “CURRENT CHALLENGES OF SCIENCE AND EDUCATION”, Berlin, Germany, July 29-31, 2024 [12].

6) XI International Scientific and Practical Conf. “SCIENCE IN THE MODERN WORLD: INNOVATIONS AND CHALLENGES”, Toronto, Canada, July 10-12, 2025 [13].

7) V International Scientific and Practical Conf. «“INTERNATIONAL EXPERIENCE IN SCIENTIFIC RESEARCH”», Chicago, USA, 18-20, December 2025 [14].

Публікація результатів дисертації. Результати десертаційної роботи опубліковано у 11 наукових працях, з яких 4 індексовані у науковометрічній базі даних Scopus, 5 одноосібних публікацій ([15]÷[18], Додаток А).

Структура і обсяг роботи. Дисертація складається зі вступу, трьох розділів, висновків, списку використаних джерел і додатків. Повний обсяг дисертації складає 229 сторінок, в тому числі: 189 сторінок основного тексту, 36 рисунків і 12 таблиць, список використаних джерел зі 134 найменувань та 7 додатків.

РОЗДІЛ 1
ОГЛЯД МЕТОДІВ КІБЕРЗАХИСТУ І УПРАВЛІННЯ ДАНИМИ В КОМП'ЮТЕРНИХ СИСТЕМАХ ТА МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ

В даному розділі наводиться стислий аналітичний огляд проблеми захисту інформації і управління розподілом даних в комп'ютерних системах та мережах у зв'язку із стрімким впровадженням мобільних мереж Інтернету речей (IoT). На основі ієрархічних класів Дж. фон Неймана і методу декомпозиції систем М. Месаровича, досліджується процес еволюції еталонних моделей Інтернету у контексті інформаційної безпеки та її сучасний стан в технологіях Інтернету речей. Аналізуються існуючі протоколи і стандарти захисту в інтерфейсах WiFi, а також моделі і методи управління цифровими потоками в мобільних ad-hoc мережах Інтернету речей.

1.1 Питання кібербезпеки в технологіях Інтернету речей (IoT)

У сучасну епоху інформаційна безпека стала однією з найактуальніших проблем і об'єктом системних досліджень, складність якої обумовлена стрімким розвитком технологій Інтернету речей, а також зростанням кількості об'єктів і суб'єктів інформаційної взаємодії у різних аспектах зберігання, доступу та передачі великих обсягів конфіденційних даних.

Дієвим інструментом дослідження об'єктів такого типу є ієрархічна декомпозиція складної проблеми на окремі відносно самостійні задачі. На сьогоднішній день, у цьому напрямку досягнуто значних результатів, зокрема, на основі теорії ієрархічних класів Дж. фон Неймана [19], методу диакоптики Г. Крона [20], теорії систем М. Месаровича [21], загальної теорії систем Л. фон Берталанфі [22], та багатьох інших, які дозволяють будувати контекстне-визначені ієрархічні моделі складних проблем. Даний підхід до ієрархічної декомпозиції об'єкта дослідження застосовується далі для системного аналізу сучасного стану проблеми кібербезпеки в технологіях Інтернету речей.

Принцип ієрархічної декомпозиції об’єкта дослідження.

Загальна ідея ієрархічної декомпозиції понять, категорій чи об’єктів відома і зрозуміла. Прикладом є універсальний десятковий класифікатор УДК для впорядкування тематики досліджень, в якому основа числення **b** =10 (кількість цифр одного розряду). Кожен розряд УДК визначає ступінь розгалуження підпорядкованих понять від 0 до 9. Будемо вважати основний об’єкт (клас) що розгалужується вниз по ієрархії таким, що має рівень 0 (верхній), тобто, він є умовно «нульовим розрядом» з одним можливим значенням: 10^0=1. Взагалі, якщо конкретний патерн УДК містить К розрядів, то максимально можлива кількість класів нижнього (К-го рівня) в ньому, очевидно, становить 10^K. Застосуємо відоме поняття «відкрита множина» (така, що містить пустий елемент ∅) для кодування ієрархічних класів. Нехай А – відкритий клас нульового (верхнього) рівня ієрархії. Тоді на першому рівні розгалуження, цифри 1÷9 визначатимуть конкретні (явно) виділені підкласи $A_{1\div9}\in A$; натомість, умовний «нульовий підклас» $A_0\in A$ будемо розуміти як «усе інше в класі А» окрім явно виділеної множини $\{A_{1\div9}\}$. У зв’язку з такою ієрархічною класифікацією (декомпозицією) понять за допомогою багато-розрядних чисел виникають щонайменше наступні питання.

Перше: яка основа системи ієрархічної класифікації понять є кращою у певному сенсі (**b**=?).

Друге: за якими критеріями обирати і виділяти класи понять і об’єктів на різних рівнях ієрархічної класифікації для аналізу складної проблеми?.

Відповідь на перше питання витікає з критерія Дж. фон Неймана для мінімізації функції «ціна апаратного відображення» деякого числа N у числовій системі з основою b: $C(b, N) = b\cdot\log_b(N) = \min$. Нескладно довести, що значення $b = e \approx 2.71828$ (основа натурального логарифму) мінімізує функцію «ціни» C(b, N). Звідси витікає, що оптимальним розгалуженням на першому рівні ієрархічної класифікації об’єктів є три гілки, тобто, ціла частина від e^1: $INT(e^1)=3$. Відповідно, на другому рівні: $INT(e^2)=7$; на третьому рівні: $INT(e^3)=20$ і так далі ([19], с. «ii», розд. 6.0 E-Elements).

Відповідь на друге питання (критерії виділення класів понять і об’єктів) дає системний метод М. Месаровича, який формалізує побудову множини відношень на заданих класах об’єктів у вигляді Декартового добутку множин.

Як відомо, останні чотири еталонні моделі Інтернету (2012, 2014, 2016, 2022) були розроблені для архітектури Інтернету речей. Серед них модель NIST SP 800-183 (2016) була відповіддю на виклик кіберзагроз, а кінцева IIC-модель (2022), разом з IIC-IIRA та IIC-IISF, формулює цілісне бачення сучасної концепції Індустрії 4.0, у тому числі, в аспекті інформаційної безпеки. З цього можна зробити висновок, що системний підхід до розв’язання складної проблеми інформаційної безпеки потребує побудови контекстно-визначеної моделі цієї проблеми згідно конкретної цілі дослідження [16].

Виходячи зі сказаного вище, для дослідження проблеми інформаційної безпеки, в дисертаційній роботі сформульовано контекстно-визначену модель S-CDM (Security-Context Defined Model) для взаємодії між клієнтами інформаційних послуг (C), інформаційними ресурсами (R) та агентами-посередниками на ринку інформаційних послуг A. При цьому, взаємодія між об’єктами C, R, A може здійснюватися як безпосередньо, так і опосередковано через інформаційно-комунікаційні мережі N, що показано на рисунку 1.1 [16].

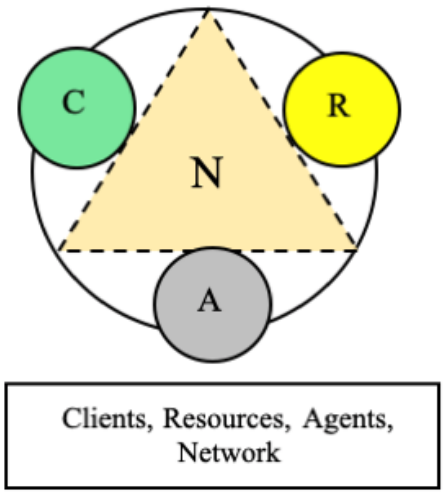


Рисунок 1.1 – Схема взаємовідносин між клієнтами C, ресурсами R та агентами A на ринку інформаційних послуг з використанням мережевих технологій N

Симетрична система відносин (1) показана матрицею S на рисунку 1.2.

S	R	C	A	N
R		1	1	1
C	1		1	1
A	1	1		1
N	1	1	1	

Рисунок 1.2 – Симетричні відносини (клієнти, ресурс, агенти) на ринку інформаційних послуг з використанням мережевих технологій

На *першому* кроці системної декомпозиції проблеми інформаційної безпеки побудуємо сімейство із трьох відношень між об'єктами R, C, A:

$$S^1:=\{(R, C), (R, A), (C, A)\}. \quad (1)$$

На *другому* кроці декомпозиції розділимо кожне з трьох відношень (1) на:

а) безпосередні відношення $(R, C)_1$, $(R, A)_2$, $(C, A)_3$;

б) опосередковані відношення між R, C, A, що здійснюються із залученням інформаційно-комунікаційних мережевих технологій, – умовно позначені індексом N від слова "Network" $(R, C)_{N4}$, $(R, A)_{N5}$, $(C, A)_{N6}$. Об'єднаємо ці два типи відношень у сімейство з 6 відношень S^2:

$$S^2:= \{[(R, C)_1, (R, A)_2, (C, A)_3], [(R, C)_{N4}, (R, A)_{N5}, (C, A)_{N6}]\}. \quad (2)$$

Припустимо, що кожен з трьох типів об'єктів (R, C, A) може бути авторизованим (справжнім, true) (R_T, C_T, A_T), або, навпаки – підробленим, фейковим (R_F, C_F, A_F).

На *третьому* кроці декомпозиції проблеми інформаційної безпеки визначимо Декартові добутки для парних взаємовідносин між об'єктами:

$$\begin{cases} (R, C)_1 \rightarrow (R_T, C_T) \times (R_F, C_F); \\ (R, A)_2 \rightarrow (R_T, A_T) \times (R_F, A_F); \\ (C, A)_3 \rightarrow (C_T, A_T) \times (C_F, A_F); \\ (R, C)_{N4} \rightarrow (R_T, C_T)_N \times (R_F, C_F)_N; \\ (R, A)_{N5} \rightarrow (R_T, A_T)_N \times (R_F, A_F)_N; \\ (C, A)_{N6} \rightarrow (C_T, A_T)_N \times (C_F, A_F)_N . \end{cases} \quad (3)$$

Схему взаємовідносин між клієнтами, ресурсами та агентами на ринку інформаційних послуг (рис.1.1), разом з відношеннями (1), (2), (3), представимо на рисунку 1.3 у вигляді ієрархічної трирівневої контекстно-визначеної моделі для комплексного аналізу проблеми інформаційної безпеки S-CDM (Security-Context Defined Model). Перший рівень моделі S-CDM (рис. 1.3) відображує три загальних типи парних відношень між ключовими об'єктами на ринку інформаційних послуг (інформаційними ресурсами R, клієнтами C та агентами-посередниками A). Другий рівень моделі відображує шість конкретизованих значень попередніх трьох типів, а саме: три безпосередні парні відношення що неявно розуміються присутніми (заради компактності рисунку при публікації відповідної статті [16] у дві колонки): $(R, C)_1$, $(R, A)_2$, $(A, C)_3$, а також три опосередковані парні відношення $(R, C)_N$, $(R, A)_N$, $(A, C)_N$, що здійснюються із залученням інформаційно-комунікаційних технологій сучасного Інтернету речей (умовно позначеного літерою N від слова "Network"). Третій рівень моделі (рис. 3) розкриває Декартові добутки парних взаємовідносин між реальними об'єктами R_T, C_T, A_T та їхніми «фейковими двійниками» R_F, C_F, A_F – на прикладі першої з шести строчок у формулі (2):

$$(R, C)_1 \rightarrow (R_T, C_T) \times (R_F, C_F) = \{(R_T, R_F), (R_T, C_F), (C_T, F_F), (C_T, C_F)\}.$$

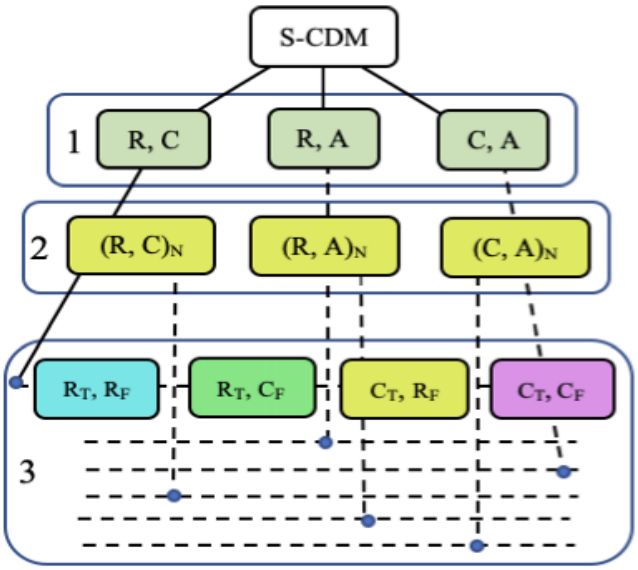


Рисунок 1.3 – Контекстно-визначена модель проблеми інформаційної безпеки S-CDM

Кількість типів відношень на третьому рівні ієрархії в моделі S-CDM (рис. 1.3) дорівнює 4×6=24. Загалом, кількість типів відношень між об'єктами R, C, A, N на трьох рівнях ієрархії в моделі S-CDM складає: перший рівень - 3; другий рівень - 6; третій рівень 24. Як бачимо, така класифікація є близькою до оптимальної класифікації Дж. фон Неймана по основі числа «e ≈ 2.71828», яка передбачає розгалуження по рівнях 1÷3 відповідно на 3, 7, 20 гілок.

Зведемо 12 типів відношень (3) у систему S^3 на рисунку 1.4. Аналогічно, 24 відношення між R, C, A, N (рис. 3) зведемо у систему S^4 на рисунку 1.5.

S^3	Тип	1	2
1	$(R, C)_1$	(R_T, C_T)	(R_F, C_F)
2	$(R, A)_2$	(R_T, A_T)	(R_F, A_F)
3	$(C, A)_3$	(C_T, A_T)	(C_F, A_F)
4	$(R, C)_{N4}$	$(R_T, C_T)_N$	$(R_F, C_F)_N$
5	$(R, A)_{N5}$	$(R_T, A_T)_N$	$(R_F, A_F)_N$
6	$(C, A)_{N6}$	$(C_T, A_T)_N$	$(C_F, A_F)_N$

Рисунок 1.4 – Таблична форма відношень між об'єктами R, C, A системи (3)

	Тип	1	2	3	4
1	$(R, C)_1$	(R_T, R_F)	(R_T, C_F)	(C_T, R_F)	(C_T, C_F)
2	$(R, A)_2$	(R_T, R_F)	(R_T, A_F)	(A_T, R_F)	(A_T, A_F)
3	$(C, A)_3$	(C_T, C_F)	(C_T, A_F)	(A_T, C_F)	(A_T, A_F)
4	$(R, C)_{N4}$	$(R_T, R_F)_{N4}$	$(R_T, C_F)_{N4}$	$(C_T, R_F)_{N4}$	$(C_T, C_F)_{N4}$
5	$(R, A)_{N5}$	$(R_T, R_F)_{N5}$	$(R_T, A_F)_{N5}$	$(A_T, R_F)_{N5}$	$(A_T, A_F)_{N5}$
6	$(C, A)_{N6}$	$(C_T, C_F)_{N5}$	$(C_T, A_F)_{N5}$	$(A_T, C_F)_{N5}$	$(A_T, A_F)_{N5}$

Рисунок 1.5 – Таблична форма відношень між об'єктами R, C, A, N

Табличні форми на рисунках 1.4 і 1.5 представляють собою два (з багатьох інших можливих) способів ієрархічної декомпозиції (класифікації) окремих задач в межах комплексної проблеми інформаційної безпеки у контексті застосування технологій Інтернету речей. Перша форма (рис. 1.4) виділяє 12 типів складових задач; друга форма з 24 типів складових задач є найбільш детально структурованою. В даній дисертаційній роботі сфокусуємося на типі $(R_T, C_T)_N$ (рис. 1.4), а саме, на процесах взаємодії між справжніми клієнтами C_T та інформаційними ресурсами R_T через мережні технології Інтернету речей N.

Відповідно до сформульованої вище моделі S-CDM, а також фокусу даної роботи на взаємодії між клієнтами ринку інформаційних послуг та власне інформаційними ресурсами опосередковано через телекомунікаційні мережі, розглянемо проблему кібербезпеки у контексті впровадження технологій сенсорних мереж та зростання сегменту IoT у загальній інфраструктурі світової Мережі [23].

Провідну роль у створенні технологій IoT у концепції промисловості майбутнього (“Industry 4.0”) відіграла Німеччина. У 1980-і роки почалась розбудова т. зв. польової шини (FieldBus) з власними послідовними протоколами від деяких німецьких виробників, спрямованими на спрощення інсталяції та обслуговування промислового обладнання. Це була мережева система розподіленого керування в режимі реального часу для підключення приладів на виробничому підприємстві.

Польова шина працює в мережах різної топології (ланцюгове з'єднання, зірка, кільце, гілка, дерево тощо). У 1987 році було запущено польову шину процесів (ProfiBus) в рамках проекту, що фінансується державою, компаніями та інститутами [24].

На основі цих досягнень, у 1991 році було розроблено одну з перших технологій Інтернету речей M-Bus для дистанційного зчитування показників лічильників комунальних послуг води, газу, електроенергії та тепла (Х. Ціглер, Університет Падерборна, разом з Texas Instruments GmbH та Techem GmbH). Стандарт M-Bus був спочатку виданий комітетом ЄС CEN у специфікації EN1434 (2002) для лічильників тепла, а потім у специфікаціях EN13757x: EN 13757-1 (2002-2022) – загальні описи для систем зв'язку; EN 13757-2 (2004-2018) – дротові рівні OSI PHY та DL; EN 13757-3 (2005-2018) – спеціалізований рівень OSI APP. Фізичний рівень підтримує асинхронний інтерфейс гальванічного типу на 2-жильному мідному кабелі. У 2003 році було розгорнуто бездротову шину M-Bus (EN 13757-4, 2008-2019) [25].

У 1998 році, альянс CSA (Connectivity Standards Alliance, раніше Zigbee Alliance), невдовзі після появи бездротових технологій Bluetooth (1994) та WiFi (1997), створив технологію ZigBee для автоматизованого збору даних з IoT пристроїв на відстані (10-100)м на частотах ISM (Industrial, Scientific, Medical)-(0.900/0.868/2.4)Ггц. ZigBee має низьке енергоспоживання і невелику швидкість передачі (20-250 Кбіт/с), натомість високий рівень захисту даних (AES-128). Ця технологія набула поширення лише у 2002 році, коли до альянсу Zigbee приєдналося більше зацікавлених компаній. Стандарт технології ZigBee (IEEE 802.15.4) був прийнятий у 2003 році та ратифікований альянсом ZigBee у 2004 році. Потім, його було перетворено у т. зв. бібліотеку Zigbee_Light_Link (2006) та версію ZigBee Pro/2007 [26].

У 2012 році опубліковано еталонну 4-рівневу модель ITU (Y.2060) для Інтернету речей [27]. У 2014 році IoT World forum прийняв еталонну 7-рівневу архітектуру Інтернету речей [28]-[29]. У червні 2016 вийшов документ ISO 20922 протоколу передачі телеметрії з чергою повідомлень (MQTT), яка стала фактичним стандартом для концепції Індустрії 4.0 – проекти модернізації виробництва та цифрова трансформація галузі, прийняття рішень у режимі реального часу, підвищення продуктивності та гнучкості [30].

У липні 2016 року інститут стандартів і технологій США NIST опублікував еталонну 4-рівневу модель IoT (NIST SP 800-183), в якій особлива увага приділялась питанням безпеки Інтернету речей [31]. Стандарт MQTT, разом з IoT-архітектурою NIST SP 800-183, відкрив новий етап «Промислового IoT» (IIoT) як ключового компоненту Індустрії 4.0, що почалася у 1968 р. з програмованого логічного контролера Р. Морлі для виробництві автоматичних трансмісій у компанії General Motors. Наразі «IIoT» розширює парадигму IoT на промислові додатки через IP-з'єднання до Інтернету для інтеграції сенсорів та обладнання з процесами збору, аналізу та обробки великих обсягів різнорідних даних. Такий підхід дозволяє приймати рішення в режимі реального часу та проводити прогнозовану аналітику, спрямовану на підвищення операційної ефективності та покращення якості продукції.

Відповідно до цього, у 2017 році стандарт 802.15.4 ZigBee був розширений до версії, що працює через Інтернет протокол IP через бібліотеку Dotdot (//:) ([[32], [33]); його поточна редакція (2020) підтримує бездротову персональну мережу (LR-WPAN) з топологією ad-hoc mesh [34].

У 2021 році, консорціум “Industry IoT Consortium” (також відомий як Industry Internet Consortium IIC™ (нині інтегрований у Digital Twin Consortium DTC™), опублікував стандарти Промислового Інтернету речей [35], а у 2022 році − відповідну структуру інтерфейсів, “The Industrial IoT Connectivity Framework - IIC-model”, де сумісність IIoT представлена 3 рівнями (технічні кінцеві пристрої, синтаксична платформа IIoT, семантичні користувачі людина/штучний інтелект). У цьому документі також представлена сучасна еталонна 3-рівнева модель Інтернету речей [36]. Однією з найбільш популярних сьогодні веб-платформ для розробки клієнтських додатків і систем IIoT є віртуальний брокер EMQX компанії EMQ, який підтримує протоколи MQTT (3.1, 3.1.1 та 5.0), HTTP, QUIC та WebSocket для майже 100 мільйонів одночасних пристроїв Інтернету речей на кластер з одним мільйоном повідомлень за секунду та затримкою в одну мілісекунду [37].

Проаналізуємо деякі аспекти проблеми кібербезпеки у контексті еволюції архітектури світової комп’ютерної Мережі від початкового Інтернету 80-х років минулого століття до сучасного індустріального Інтернету речей.

Функціональну структуру комп’ютерних мереж, зазвичай, відображують моделями апаратно-програмних мережевих інтерфейсів, пристроїв і процесів в них, а представлення, кодування і захист інформації обумовлені рівнем відповідної моделі, типом інтерфейсу та протоколами міжрівневої взаємодії.

За фактом, світова мережа Інтернет, значною мірою, спирається на 4-рівневу архітектуру стека протоколів TCP/IP (1983) проекту ARPANET для Міноборони США. Натомість, популярна 7-рівнева еталонна модель OSI (1984) Міжнародної організації по стандартизації ISO концептуально описує термінологію взаємодії т. зв. “відкритих систем” на абстрактних рівнях ієрархії.

Моделі OSI та TCP/IP певний час розвивались паралельно, однак у намаганні вичерпного опису мережі, модель OSI/ISO не отримала підтримку розробників Інтернету. З іншого боку, поява додатків реального часу виявила обмеження моделі TCP/IP, переважно орієнтованої на передачу файлових даних. У 2004 році Міжнародний союз електрозв'язку (ITU) анонсував 3-рівневу модель мереж наступних поколінь NGN для підтримки нових вимог до якості обслуговування. Стрімке поширення сенсорних мереж IoT створили потужні потоки різноманітних типів даних непритаманних мережам TCP/IP, внаслідок чого, три попередні моделі (OSI, TCP/IP, NGN) послідовно доповнили чотири нові моделі Інтернету речей від різних міжнародних організацій:

- 2012: модель IoT/ITU Y.2060 (4 рівні: пристрої, мережа, сервіси, додатки);

- 2014: модель IoT/IoTWF World Forum Reference Model (7 рівнів: пристрої, інтерфейси, туманні обчислення, акумуляція даних, агрегація даних, додатки, бізнес-процеси);

- 2016: модель NIST SP 800-183 (4 рівні: сенсори, агрегатори, комунікації, управління; ця модель сфокусована на безпеці комп'ютерних систем);

- 2022: модель Industrial IoT (IIoT/IIC^{TM}) Reference Architecture (3 рівні: кінцеві пристрої, IIoT- платформа, людські та цифрові користувачі).

Вищезгадана модель IoT/ITU Y.2060 (2012) визначає 4 рівні мережевих інтерфейсів: пристрої, мережевий транспорт, сервісну платформу та додатки. Задекларовано два види можливостей безпеки: загальні (авторизація, автентифікація, захист цілісності даних, контроль доступу тощо) та специфічні для Інтернету речей опції (без конкретних деталей).

Наступна модель IoT/IoTWF (2014) визначає 7 рівнів: (пристрої IoT, підключення, граничні хмарні обчислення, накопичення даних, агрегація даних, додатки, бізнес-процеси). У ті роки спостерігалося значне зростання сегмента IoT, тому ця модель в основному спрямована на технологічні аспекти управління великими даними. Конфіденційність IoT тоді не привертала достатньої уваги, в результаті чого виникли серйозні проблеми.

Для вирішення нових проблем конфіденційності IoT, NIST у 2016 році представив еталонну модель (SP 800-183) що визначає 4 рівні конфіденційності: 1) кластери датчиків з індивідуальними вагами впливу d_i; 2) агрегатори підсумовування Sum(di); 3) канали зв'язку (USB, бездротові, дротові, вербальні тощо), включно з тими що пропонуються як послуга AAS (As a Service); 4) зовнішні утиліти AAS (eUtilities), де людина може розглядатися як послуга eUtility. Дані від eUtility мають ваги, а утиліта eUtility, що не є людиною, може мати ідентифікатор пристрою, що є вирішальним для ідентифікації/автентифікації. Ключову роль у цій моделі відіграють так звані «тригери прийняття рішень» (DT), що діють на будь-якому рівні та формалізують основну мету Інтернету речей на верхньому рівні (eUtility). Структура мережі IoT за цією моделлю подібна до просторово розподіленої штучної нейронної мережі. Таким чином, складна системна проблема кібербезпеки IoT функціонально розкладається на чотири окремі шари відповідно до визначених 4 рівнів моделі. Діюча 3-рівнева модель IIoT/IIC™ (2022) охоплює «Еталонну архітектуру промислового Інтернету» IIC-IIRA [38] та «Структуру промислової інтернет-безпеки» IIC-IISF [39]. Наразі, ці три документи найбільш системно описують загальну архітектуру IIoT і концепцію кібербезпеки в сучасному цифровому світі.

Модель IIC може бути інтерпретована в термінах OSI та TCP/IP, а саме. Рівні OSI L1÷L4 об'єднаємо в транспортний рівень T для доставки повідомлень TCP/UDP – назвемо це «технічна сумісність TchIO (Technical Inter-Operability). Рівні L5÷L7 об'єднаємо та структуруємо наступним чином: фрейм даних F (файл стану або подій що сформований повідомленнями) – назвемо це «синтаксична сумісність SinIO (Syntax Inter-Operability); документ D (Сумісність типу SinIO означає можливість обміну інформації в рамках спільної структури даних (наприклад, у відомій граматиці мови), натомість, сумісність типу SemIO – це можливість однозначної контекстне-обумовленої інтерпретації інформаційних даних.

Модель IIC-IIRA (2022) складається з 3 рівнів: 1) edge (мережа доступу до Інтернету речей); 2) інтернет-платформа послуг та промисловий програмний інтерфейс API або людина ([38], с. 44). Вона близька за своєю суттю до згаданої вище 4-рівневої моделі NIST SP 800-183, якщо комунікації поєднати з платформою послуг. Така структурна інтерпретація розділяє типовий IoT-домен на три складові частини (edge, платформа, API) з трьома типами взаємодії: «edge-платформа», «edge-API» та «платформа-API», які включають внутрішні та зовнішні аспекти безпеки.

Модель IIC-IISF (2016) надає рекомендації щодо надійних систем, характеристик безпеки, технологій та методів, що застосовуються, а також щодо того, як отримати впевненість у тому, що відповідний набір питань було вирішено для задоволення очікувань зацікавлених сторін. Основний термін «Надійність системи IIoT» визначається як сукупність 5 сутностей (стійкість, надійність, безпека, конфіденційність, захищеність) що піддаються чотирьом зовнішнім впливам: порушенням навколишнього середовища, системним збоям, атакам, людським помилкам ([39], с. 23).

Важливо зазначити, що кожна нова модель мережевої архітектури не відкидала попередні, а навпаки, доповнювала та розвивала їх у контексті нових конкретних завдань. Тому всі ці моделі, так чи інакше, продовжують використовуватися фахівцями. Семирівнева еталонна модель OSI/ISO визначає загальноприйняті професійні терміни абстрактної відкритої системи, тоді як чотирирівневий стек TCP/IP формулює детальні інтерфейси взаємодії реальних мережевих об'єктів. Трирівнева модель NGN/ITU відображає будь-яку відкриту пакетну мережу (включаючи весь Інтернет) у досить простому вигляді: транспортна інфраструктура (нижній рівень) та прикладна область (верхній рівень), які взаємодіють через IP-протокол (середній рівень).

Останні чотири моделі (2012, 2014, 2016, 2022) були розроблені для архітектури IoT-мереж; серед них модель NIST SP 800-183 (2016) була відповіддю на виклик кібербезпеки, а кінцева IIC-модель (2022) разом з IIC-IIRA та IIC-IISF формують цілісне бачення сучасної концепції Індустрії 4.0.

У контексті завдань дисертаційного дослідження, підсумуємо основні положення концепції Індустріального Інтернету речей IIoT стосовно кібербезпеки, які сформульовані у розділі “Security Framework” відповідного документа [39].

Відбувається конвергенція інформаційних і операційних технологій (с. 24). Змінюються підходи щодо інформаційної та операційної безпеки (с. 25). Запроваджуються нові рішення поверх застарілої інфраструктури операційних технологій ("brownfield (new + legacy) deployments in OT"), що є найбільш поширеним, але водночас, найскладнішим сценарій для забезпечення надійного кіберзахисту (с. 26). Важаються критичною точкою вразливості хмарні обчислення, оскільки використання спільних сторонніх постачальників послуг створює низку обмежень довіри, що впливають на безпеку та конфіденційність (с. 27). Розшаровуються високо-рівневі вимоги, моделі і політики безпеки на детальні, конкретні та керовані функції та політики (с. 59).

Останній принцип (розшарування високо-рівневих вимог у процесі розробки безпечних IoT-систем) спрямований на перехід від абстрактних цілей до практичної реалізації. Зокрема, фреймворк IISF визначає, що спершу потрібно встановити загальний «рівень довіри» (trustworthiness), який включає безпеку, надійність, стійкість і конфіденційність. Потім, за допомогою «функціонального розшарування», ці загальні цілі розділяються на конкретні «функції безпеки» (наприклад, автентифікація, авторизація, шифрування та моніторинг). На основі цих функцій розробляються детальні «політики» (набори правил і процедур, які реалізують кожну функцію). Наприклад, політика може визначати, що усі користувачі мають застосовувати двофакторну автентифікацію. Таким чином, довільна узагальнена модель безпеки та політики не розглядаються як моноліт, натомість стають результатом системного аналізу та розкладання на менш складні, керовані компоненти. Це дозволяє створювати надійну та цілісну архітектуру безпеки для комплексних промислових систем IoT.

Огляд існуючих на сьогодні семи еталонних моделей Інтернету (TCP/IP-1983, OSI-1984, NGN-2004, IoT/Y.2060-2012, IoT/IoTWF-2014, NIST SP 800-183/2016, IIoT-2022), а також сучасних тенденцій розвитку Інтернету речей, свідчать про незавершеність архітектури Інтернету в цілому, що обумовлює насущну потребу у подальших дослідженнях в цьому напрямку. Зокрема, фахівці вважають проблеми сумісності (interoperability) та інформаційної безпеки сучасних мереж Інтернету речей одними з ключових викликів нашого 21-го століття [40].

Мережі IoT оперують великими обсягами конфіденційних даних, починаючи від особистої інформації і закінчуючи важливими бізнес-даними. Багато сегментів IoT вбудовані в критичну інфраструктуру, таку як енергомережі, водопостачання та транспорт. Системи і мережі IoT широко використовують бездротові інтерфейси і канали зв’язку що є найбільш вразливими ланками їхньої безпеки [41].

Так, стандарт ZigBee дозволяє використання ключів з'єднання для повторного приєднання до мережі; отже, зловмисник може клонувати легітимний пристрій і підробити мережевий рівень CA (Certification Authority), видаючи себе за раніше підключений пристрій, який хотів повторно приєднатися до мережі [42]. Іншими ланками вразливості IoT є програмне забезпечення контролерів, пристрої розподілу трафіку [43].

Файли типу “exe” та “com” можуть містити вірусні програмні закладки (bookmarks), які самі по собі не є носіями вірусу, натомість посилаються на Web-ресурс, що його містить. PHP-програми Web-серверів здатні приховувати так звані “чорні двері” (backdoors) – зловмисний програмний код, що скасовує нормальні процедури автентифікації доступу; в результаті, зловмисник отримує віддалений контроль інформаційних ресурсів [37].

Витоки інформації та шкідливі втручання можуть здійснюватися прихованими каналами взаємодії через вірусні коди у файлах типу “jpeg”. Детальний огляд потенційних загроз в мережах IoT і розширена бібліографія наведені в [44].

Новітні моделі Інтернету речей (NIST SP 800-183 та IIoT) розглядають радіоканали зв'язку як критичний елемент ланцюга безпеки. Систематизуємо ці канали у контексті управління мобільними ad-hoc мережами, таблиця 1.1.

Таблиця 1.1 – Радіоканали мобільних ad-hoc мереж

Відстань (км)	Тип каналу зв'язку	Тип інтерфейсу взаємодії	Протокол
1.5÷2 км	Радіо 2.4 ГГц (на прямій видимості) DSSS/FHSS модуляція	DSSS (Direct Sequence Spread Spectrum) + FHSS (Frequency Hopping 23 канали)	**DSM2** / **DSMX** (Закритий) Вибір 2-х вільних частот/перескакування на вільні канали (один з 23-х)
3÷10 км	Оптоволоконний 0.700 ÷ 1.100 ГГц	**FPV** (First Person Vision)	Закритий протокол **TBS** Crossfire від компанії Team Black Sheep (Hong Kong)
5÷35 км	Радіо 2.405-2479 ГГц (програмна регуляція)	**SDI** (Serial Digital Interface) телеметрія + **HDMI** (High-Definition Multimedia Interface) Full HD video (продукт наприбуткової організації Dronecode Foundation/Linux)	**MAVLink** (Micro Air Vehicle Link) Відкритий легкий протокол L7 (C++/Python)(ArduPilot, PX4 Autopilot) Контроль цілісності, маршрутизація
10÷50 км	Радіо 868/915 МГц та 2.4 ГГц (на прямій видимості)	Micro-LoRa (Long Range модуляція) від "ExpressLRS LLC&Community" - група провідних розробників (як юридична оболонка), незалежна від комерційних гігантів.	ExpressLRS -відкритий легкий протокол L7 (C++/Python) для пакетів телеметрії з малою затрим. на швидкості (1000+ пакетів/с). (ArduPilot, PX4 Autopilot)
50÷100+км	Мобільні мережі (LTE/5G) з OFDM модуляцією	UDP-інкапсуляція MAVLink: Найпростіший і найшвидший спосіб передачі даних через мобільний інтернет.	SRT (Secure Reliable Transport)-більше для відео, але також для передачі телеметрії в «одному пакеті» з відеопотоком через 4G/5G.
	Супутниковий зв'язок. Starlink	Iridium SBD (Short Burst Data): Затримка велика (секунди), але працює всюди.	Протокол для передачі коротких пакетів телеметрії через супутники Iridium.
	Військові системи	STANAG 4586 (стандарт НАТО)	Спеціалізовані **SDR**-протоколи (Software Defined Radio).

1.2 Методи захисту інформації в системах та мережах IoT

В комп'ютерних системах та мережах Інтернету речей на різних рівнях ієрархії діють специфічні підходи до захисту інформації від несанкціонованого доступу (НСД). Так, на фізичному рівні OSI використовують методи кодування, які не тільки збільшують завадостійкість, але й водночас, підвищують захист від НСД на основі спеціальних сигнально-кодових конструкцій (СКК) на кшталт шумо-подібних таймерних сигнальних конструкцій (ТСК). На канальному рівні і вище широко застосовують різноманітні методи криптографічного кодування інформації.

Мережеві додатки Інтернету речей, як правило, мають архітектуру типу «Клієнт-сервер». Базовими принципами кібербезпеки сучасної клієнт-серверної мережі є *конфіденційність*, *цілісність* та *доступність*, відомі як тріада CIA (confidentiality, integrity, availability). Конфіденційність – це обмежений (авторизований) доступ до ресурсів мережі (напр., за механізмом «логін-пароль»). Цілісність – це гарантія того, що інформація не була змінена без дозволу. Доступність – це можливість доступу авторизованих користувачів до ресурсів мережі тоді, коли це потрібно (зокрема, захист від збоїв) [45], [46].

Базовими компонентами захисту даних в мережі типу «Клієнт-сервер» є *ідентифікація*, *автентифікація* та *авторизація* клієнтів мережі.

Ідентифікація – це підтвердження права доступу клієнта до мережі через особистий унікальний ідентифікатор (ID), напр., комбінацію літер та/або цифр, яка попередньо зареєстрована на сервері у списку клієнтів мережі. В мережі WiFi клієнтським ID є MAC-адреса WiFi інтерфейсу (6 байт).

Автентифікація – це підтвердження того, що клієнт ID насправді є тим, за кого себе видає; реалізується через унікальні ознаки ідентичності клієнта, що фіксуються на сервері в процесі реєстрації клієнта (пароль доступу, особісті дані, відбитки пальців тощо).

Авторизація – це перевірка прав клієнта ID на доступ до ресурсів мережі та операції з ними згідно переліку прав даного клієнта, зафіксованих на сервері при його реєстрації.

Розглянемо методи захисту інформації в системах та мережах Інтернету речей у контексті протоколів і стандартів безпеки для однієї з найбільш популярних бездротових технологій WiFi. Виділимо трійку компонентів у протоколах безпеки WiFi: алгоритм шифрування (шифр, сіпфер) метод автентифікації, структура фрейму (крипто-протокол), – таблиця 2 [47].

Таблиця 1.2 – Протоколи безпеки технології WiFi

Протокол безпеки	Шифр	Автентифікація	Крипто-протокол
WEP (1997)	RC4	SKA	WEP-Encapsulation
WPA (2003)	RC4	PSK (4-Way Handshake) /EAP 802.1X- RADIUS	TKIP
WPA2 (2004)	AES-128 bit	PSK	CCMP
WPA3 (2018)	AES-128/256 bit	SAE (Dragonfly)	CCMP/GCMP
Wi-Fi 7 (802.11be)	EHT (Extremely High Throughput) - багато-смугова супер-магістраль (перспективний стандарт)		

1) Шифр (cipher, сіпфер) – алгоритм шифрування деякої впорядкованої множини символів у двійковій чи іншій позиційній системі. Наприклад: RC4 у стандартах WEP та WPA; Salsa20, ChaCha20, DES; AES (Advanced Encryption Standard) у WPA2 та WPA3; RSA (Rivest–Shamir–Adleman); ECC (Elliptic-Curve Cryptography) – криптографія на еліптичних кривих; алгоритм цифрового підпису DSA (Digital Signature Algorithm).

2) Автентифікація – метод обміну ключами шифрування між взаємодіючими сторонами. Наприклад: SKA (Shared Key Authentication) у WEP; EAP (Extensible Authentication Protocol) у WPA; PSK (Pre-Shared Key) у WPA2; SAE (Simultaneous Authentication of Equals) у WPA3.

3) Крипто-протокол – спосіб побудови фреймів для захищеної взаємодії клієнтів мережі через канал зв’язку на основі деякого алгоритму шифрування і методу автентифікації. Наприклад: “WEP-Encapsulation” у стандарті WEP/WiFi; послідовний крипто-протокол CCMP (Counter Cipher Message Protocol) у WPA2/WiFi; швидкий паралельний крипто-протокол GCMP (Galois Counter Mode Protocol) у WPA3/WiF.

Алгоритми шифрування даних в системах та мережах Інтернету речей

Розглянемо особливості алгоритмів шифрування даних в системах та мережах Інтернету речей відповідно до виділеної вище трійки компонентів для протоколів (стандартів) безпеки WiFi (шифр, автентифікація, крипто-протокол, - див. Таб. 2). На рисунку 1.6 показано ієрархічно впорядковані класичні алгоритми симетричного та асиметричного шифрування. Вони відрізняються структурою, стійкістю та сферою використання [48], [49], [50], [51].

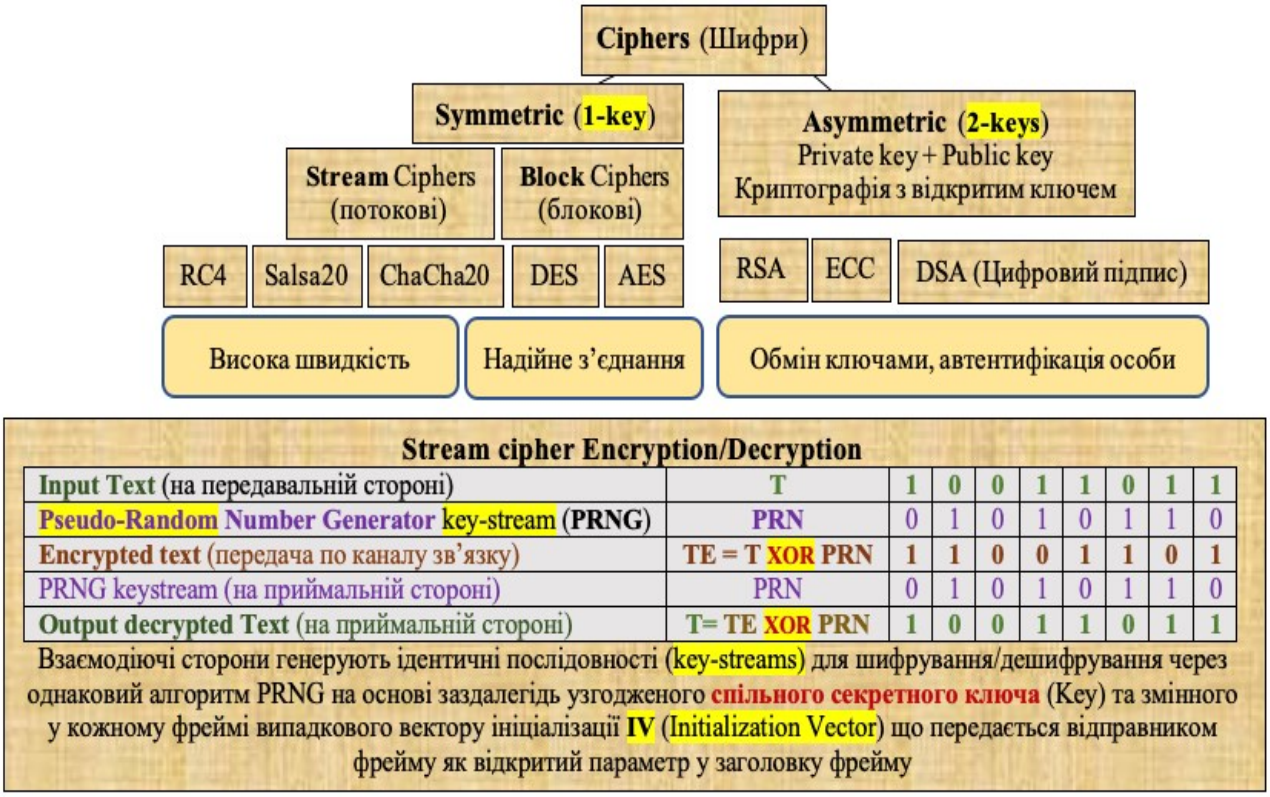


Рисунок 1.6 – Класичні алгоритми симетричного та асиметричного шифрування

Симетричні шифри зазвичай розділяють на потокові (stream ciphers) та блокові (block ciphers), особливості яких представлено на рисунку 7. Типовий симетричний потоковий шифр оперує одним (симетричним) ключом для послідовного шифрування/дешифрування окремих символів вхідного потоку, структура (текст T) якого формується передавальною стороною на основі полів фрейму згідно конкретного крипто-протоколу (напр., WEP).

Для шифрування, символи тексту T по-бітах склада.ться по модулю 2 (операція XOR) з псевдо-випадковим потоком біт PRN (Pseudo-Random Numbers) від генератора псевдовипадкових чисел PRNG (PRN Generator) на основі секретного ключа Key і відкритого вектора ініціалізації IV (Intialization Vector). Результатом є крипто-текст TE=T**XOR**PRN, вкладений у фрейм разом з вектором ініціалізації IV для передачі по каналу. На приймальній стороні, генератор PRNG формує такий самий потік PRN на основі ідентичного ключа Key та вектора IV. Повторне застосування операції XOR до шифро-тексту TE перетворює його в оригінальний текст T (є інверсним оператором I):

TE**XOR**PRN= (T**XOR**PRN)**XOR**PRN = T[(**XOR**PRN)(**XOR**PRN)] = T× I = T.

Симетричні потокові шифри є простими і швидкими у програмній та апаратній реалізації, що є їхньою головною первагою. Тому їх застосовують там, де потрібна висока швидкість і дані постійно надходять (потокове відео, VoIP) – наприклад, шифр ChaCha20. Натомість, вони вимагають точної логічної синхронізації псевдо-випадкових послідовностей PRN між відправником і одержувачем (якщо ця синхронізація втрачається, то дешифрування стає неможливим). Окрім того, вони чутливі до повторного використання ключа і вектора ініціалізації (тобто однакового потоку PRN для шифрування різних фреймів), оскільки це суттєво зменшує стійкість шифру до зламу. Врахування і зменшення впливу цих недоліків потокових шифрів у протоколах безпеки є нетривіальною технічною задачею.

Сучасні протоколи безпеки вимагають також надійної автентифікації і контроля цілісності даних (незмінність під час передачі). Для шифрування даних на диску або у захищеному з’єднанні, використовують симетричні блокові шифри. Наприклад, шифр AES-128 в режимі GCM (Galois/Counter Mode), який опідтримує *конфіденційність* інформації та *автентифікацію* її джерела. Дані шифруються в потоковому режимі (Counter Mode). Водночас, створюється крипто-код автентифікації в полі Галуа (Galois field) як гарантію того, що дані не були підроблені. Режим GCM підтримує пропускну здатність сучасних швидкісних каналів зв’язку недорогими апаратними ресурсами [52].

Для безпечного обміну ключами шифрування або підтвердження автентичності особи (електронний підпис), зазвичай, використовують *асиметричні шифри* з двома ключами (, такі як (RSA (R. Rivest, A. Shamir, L. Adleman, MIT, USA, 1977), ECC (Elliptic Curve Cryptography, N. Koblitz, V. Miller, USA, 1985) [53], [54].

Приватний (секретний) ключ (private key) відомий лише власнику (тобто, отримувачу шифрованого фрейме). Відкритий (публічний) ключ (public key) надається власником іншим взаємодіючим сторонам через канал зв'яку в якості параметра фрейму для шифрування наступних фреймів. Зашифровані відкритим (публічним). ключом фрейми може дешифрувати тільки власник приватного (секретного) ключа.

Даний тип шифрування також називають "Public Key Cryptography" (PKC). Особливості методів симетричного та асиметричного шифрування відображено на рисунку 1.7.

I. Симетричний шифр (Symmetric Cipher). Шифрування/дешифрування одним секретним ключем.

Тип Шифру	Особливість	Приклади
Потоковий Шифр (Stream Cipher)	Працює з даними послідовно по бітах/символах. Простий і значно швидший за блокові шифри.	**RC4** - застарілий **Salsa20, ChaCha20** - сучасні.
Блоковий Шифр (Block Cipher)	Оперує фіксованими блоками даних 128/256 біт. Кожен блок шифрується незалежно. Використовується для шифрування великих обсягів даних на диску, а також для створення високо захищеного з'єднання.	**DES** - застарілий; **AES** (Advanced Encryption Standard) - сучасний.

II. Асиметричний шифр (Asymmetric Cipher). Шифрування/дешифрування двома різними ключами
Відкритий ключ (Public Key): Використовується для шифрування даних або перевірки підпису.
Приватний ключ (Private Key): Відомий лише власнику. Для дешифрування даних або створення підпису.

Тип Шифру	Особливість	Приклади
Криптографія з відкритим ключем (Public Key Cryptography PKC)	Використовується для обміну ключами, цифрових підписів та захищеної передачі даних, коли сторони не зустрічалися. Математично складніший і значно повільніший за симетричні шифри.	**RSA**, **ECC** (Еліптичні криві), **DSA** (для підпису).

Рисунок 1.7 – Особливості симетричного та асиметричного шифрів

Розглянемо типові алгоритми шифрування (шифри, сіпфери).

Rivest cipher RC4. Симетричний потоковий шифр (R.L. Rivest, 1987). Є складовою частиною протоколу шифрування WEP (одного з перших в технології WiFi), який досі підтримується багатьма пристроями. Нині сіпфер RC4 вважається застарілим і небезпечним для використання в нових системах, натомість, є важливою віхою еволюції криптографії, потенціал якого ще не вичерпано, і представляє інтерес для вирішення окремих спеціальних задач.

Сіпфер RC4 реалізує наступні три фази для кожного заданого ключа шифрування **K** розміром 1÷256 байтових символів (зазвичай, 16/32 символи).

1) Ініціалізація вектора стану $\mathbf{S_K}$ з 256 байтових символів по алгоритму перемішування символів KSA (Key-Scheduling Algorithm). Для цього, вектор стану S спочатку заповнюється числами від 0 до 255. Окрім того, створюється тимчасовий вектор $\mathbf{T_K}$ такого ж розміру шляхом його циклічного заповнення символами ключа K. Далі, в циклі по індексу «i», кожному символу S(i) знаходиться індекс $j \in [0, 255]$ для перестановки символів за правилом:

$$j = [i + S(i) + T(i)] \bmod 256;$$

$S(i) \leftrightarrow S(j)$. Результатом є вектор стану $\mathbf{S_K}$.

2) Генерація чергових байтових символів **Z**(t) ключового потоку {**Z**} для кожного вхідного байтового символу **P**(t) на основі вектора стану $\mathbf{S_K}$ по алгоритму PRGA (Pseudo-Random Generation Algorithm). Для цього спочатку обнуляються індекси «i», «j»: i = 0; j = 0. Далі у відкритому циклі по індексу «i» здійснюються наступні операції:

$i = (i + 1)/mod\_256;$

$j= [(j+S(i)]/mod\_256;$

$S(i) \leftrightarrow S(j)$ - перестановка місцями;

$t = [S(i) + S(j)]/mod\_256;$

$Z = S(t)$ - черговий байтовий символ ключового потоку.

3) Шифрування чергового вхідного байтового символу **P** через по-бітову операцію $\oplus$ складання з символом **Z** по модулю 2 (операція XOR):

$$P \rightarrow C=(P \oplus Z).$$

Завдяки швидкості і простоті, сіпфер RC4 був популярним в сокетах SSL/TSL мереж WiFi, однак через свої вразливості, зараз не використовується для критичних застосувань. Сіпфер RC4 пройшов наступні фази еволюції і зміни назви [55].

- SPRITZ – шифр створений для збільшення рівня захисту RC4 (наразі, SPRITZ не використовується у WiFi). На основі ключа і короткого блоку даних генерує псевдовипадковий «ключовий потік біт» що складається по модулю 2 (XOR) з вхідними даними для шифрування. Тим же ключом генерується «ключовий потік» для дешифрування даних.

- RC4A, VMPC (Variable Modified Permutation Composition). Підвищують швидкість і надійність, але має недостатню випадковість генератора чисел.

- RC4A+ вдосконалює RC4A, але ціною збільшення своє складності.

Типовим для сіпфера RC4 є шифрування з 64/128/256-бітними ключами. Однак, у RC4 було виявлено недолік, через який 128-бітний ключ можна було зламати за секунди.

Іншу вразливість RC4 було виявлено при його використанні у ланцюгах блокчейн-шифрування (знайдено спосіб обійти RC4 лише з незначним збільшенням обчислювальної потужності, необхідної в попередній атаці на RC4). Зараз RC4 вважається застарілим по своїх параметрах захищеності, натомість його алгоритм формування шифрованих фреймів є типовим, і застосовується у інших сіпферах.

Симетричний блоковий шифр DES (Data Encryption Standard, FIPS, USA, 1977). Шифр DES оперує блоками даних 64 біт з ключом 56 біт. В основі алгоритму лежить структура “Feistel Network” (FN) що циклічно виконує серії підстановок та перестановок.

Через малий розмір ключа, DES є нестійким до атак повного перебору (brute-force attacks), і на сучасних обчислювальних машинах його можна зламати за лічені години. Сьогодні стандарт DES вважається застарілим і практично не використовується [56].

Симетричний блоковий шифр AES (Advanced Encryption Standard, NIST, USA, 2001). Шифр, оперує блоками даних розміром 128 біти з ключами 128/192/256 біт. В основі AES лежить структура заміщення-перестановки “Substitution-Permutation Network” (SPN), яка більш ефективна порівняно зі структурою “Feistel Network” (FN) у алгоритмі шифрування DES. Шифр AES, завдяки більшого розміру ключа, є стійким до атак повного перебору. Вважається, що навіть на суперкомп’ютері, для зламу шифру AES-128 біт потрібні мільярди років.

Наразі, AES є діючим світовим стандартом шифрування, рекомендованим для захисту даних у більшості сучасних систем, від WiFi до фінансових операцій. Однак, стійкість AES має свою зворотну сторону – низьку продуктивність і високі вимоги до ресурсів порівняно з іншими алгоритмами, що може бути критичним для вирішення певних задач.

Причиною високої затримки (Latency) шифру AES з блоковим шифром є складні операції (зокрема, S-box_lookup та MixColumns, яка вимагає множення у полі Галуа). Для комп’ютерних систем, що потребують високої пропускної здатності або працюють на обмежених ресурсах (наприклад, IoT-пристрої, потокове відео на слабких процесорах), шифр AES є повільнішим сучасних потокових шифрів (так, шифр ChaCha20 використовує прості операції XOR і циклічний зсув, які швидше реалізуються програмно). Також, AES-128 має вищі витрати ресурсів (кількість циклів на обробку одного байту даних), оскільки для високої криптографічної стійкості вимагає 10 раундів обробки інформації.

Асиметричний шифр RSA (R. Rivest, A. Shamir, L. Adleman, MIT, USA, 1977). Здійснює шифруваня текстових повідомлень по окремих символах на принципі факторизації великих цілих чисел (розкладання їх на прості множники). На сьогодні є одним з поширених методів крипто захисту [53].

Профіль RSA визначає трійка натуральних чисел (N, E, D), де $N = (p \cdot q)$ – добуток двох простих чисел; E – відкритий ключ шифрування; D – секретний ключ (відомий лише власнику конкретного профіля шифру).

Ключі E, D розраховується на основі функції Ейлера ϕ(N):

$$\begin{cases} N = p \cdot q; \\ \phi(N) = (p-1) \cdot (q-1); \\ 1 < E < \phi(N); \\ \text{НСД } [E, \phi(N)] = 1; \\ (D \cdot E) \text{ /mod\_}\phi(N) = 1 , \end{cases}$$

де НСД – найбільший спільний дільник; (N)/mod_M – операція «ділення натурального числа N по модулю M» (залишок від ділення N/M). Наприклад:

$$\begin{cases} p = 2;\ q = 29; \quad N = 2 \cdot 29 = 58; \\ \phi(p, q) = (2-1) \cdot (29-1) = 28; \\ E = 3; \\ (D \cdot 3) \text{ /mod\_}28 = 1. \text{ Звідси } D = \mathbf{19}. \end{cases}$$

Секретний ключ D обчислюється з останнього рівняння за допомогою розширеного алгоритму Евкліда.

Розглянемо приклад шифрування/дешифрування непечатного символу “ENQ” з десятковим значенням «**5**» у коді ASCII (команда “Inquiery” – запит до хоста на відповідь в системах передачі даних).

Шифрування: $C = (5\text{^}E)/\text{mod\_}N = (5^{\mathbf{3}})/\text{mod\_}58 = 125 – 2 \cdot 58 = 125 – 116 = 9.$

Дешифрування: $(C\text{^}D)/\text{mod\_}N = (9^{\mathbf{19}})/\text{mod\_}58 = \mathbf{5}$ (це обчислення складніше).

Оператор шифрування (Z^D)/mod_N є інверсним до оператора дешифрування (Z^E)/mod_N, і повертає значення вхідного символу Z:

$$[(Z\text{^}D)/\text{mod\_}N]\text{^}E)/\text{mod\_}N \equiv Z.$$

Асиметричний шифр ECC (Elliptic Curve Cryptography, N. Koblitz, V. Miller, USA, 1985). Використовує складність розв’язання задачі дискретного логарифма на еліптичних кривих, яка ґрунтується на властивостях точок на еліптичній кривій над скінченним полем. Ця математична проблема є складнішою для розв’язання, ніж факторизація; тому ключі ECC можуть бути коротшими, ніж ключі RSA. Наприклад, ECC ключ розміром 256 біт надає такий самий рівень захисту, як RSA ключ розміром 3072 біт [54].

Порівняємо між собою симетричні шифри RSA та ECC.

Головна перевага ECC перед RSA – висока криптографічна міцність на одиницю довжини ключа про тому ж рівні безпеки. Тому ECC є популярним для мобільних пристроїв, сервісів крипто-валют типу Bitcoin, бездротових сенсорних мереж та інших систем з обмеженими ресурсами.

Зворотна сторона шифру ECC – його значна повільність і складність у програмній реалізації, зокрема операціях додавання та множення точок еліптичної кривої в достатньо великому скінченному полі Галуа. Окрім того, деякі ранні реалізації шифру ECC закриті патентами, що створює юридичні перешкоди для впровадження відкритого програмного забезпечення, змушуючи розробників обирати «вільні» еліптичні криві, (напр., розроблені національним інститутом стандартів і технологій США NIST).

Інший недолік шифру ECC – залежність від обраної еліптичної кривої (наприклад, NIST P-256, Curve25519). Якщо виявиться слабкість у математичній структурі конкретної кривої (наприклад, через підозру, що крива була «підтасована» розробником для створення «лазівки»), це може скомпрометувати всю систему.

На відміну від шифру RSA, де безпека базується на одній добре вивченій проблемі факторизації, шифр ECC спирається на довіру, що обрана еліптична крива, як основа шифру ECC, є надійною.

Порівняльні характеристики асиметричних шифрів RSA та ECC зведено у таблицю 1.3.

Таблиця 1.3 –Характеристики асиметричних шифрів RSA та ECC

Характеристика	RSA	ECC
Математична основа	Факторизація великих чисел (Використовується, поступово замінюється на ECC)	Обчислення дискретного логарифма на еліптичних кривих (діючий стандарт)
Розмір ключа еквівалентної безпеки	Значний (2048÷4096 біт)	Значно менший (наприклад, 224-384 біт)
Продуктивність	Відносно низька	Висока (швидка)
Обмеження	Не підходить для ресурсно-обмежених пристроїв	Вигідно для мобільних та IoT-систем

Розглянемо особливості застосування асиметричних шифрів RSA та ECC у взаємодії з типовими протоколами безпеки транспортного рівня SSL і TLS.

Протокол безпеки SSL (Secure Sockets Layer) – це застарілий (1990) попередник сучасного протоколу безпеки транспортного рівня TLS (Transport Layer Security) від компанії Netscape (1990-і роки) для захищеного з'єднання між сервером і веб-браузером в мережі Інтернет. Попри те, що термін «SSL» досі широко використовується, сучасні браузери та сервери більше не підтримують SSL через виявлені в ньому вразливості. Зокрема, у SSL було знайдено серйозні недоліки на кшталт POODLE-атаки (Padding Oracle On Downgraded Legacy Encryption), які дозволяли зловмисникам перехоплювати зашифровані дані. Остання версія SSL 3.0, була офіційно визнана застарілою у 2015 році [57].

Коли сьогодні іноді зустрічається згадка про SSL-сертифікат або SSL-з'єднання, фактично йдеться про протокол TLS , який був розроблений та опублікований у версії TLS 1.0 у січні 1999 року (RFC 2246) як відповідь на вразливості SSL. Основними функціями протоколів SSL/TLS є конфіденційність (запобігання перехопленню та читанню даних), цілісність (гарантування незмінності даних під час передачі) та автентифікація (можливість клієнта переконатися у справжності сервера).

Принцип роботи крипто-протоколу TLS полягає у наступному (TLS 1.3, RFC 8446). Протокол TLS встановлює зашифроване логічне з'єднання транспортного рівня між клієнтом (напр., браузером клієнта) і сервером (веб-сайтом). Цей процес, відомий як TLS-Handshake (рукостискання TLS). Він складається з наступних кроків.

1) Client Hello (Привітання клієнта). Браузер надсилає серверу запит на встановлення захищеного з'єднання; у ньому міститься інформація про версію TLS, яку підтримує браузер, список криптографічних алгоритмів (Cipher Suites).

2) Server Hello (Привітання сервера). Сервер відповідає, обираючи найвищу спільну версію TLS і найкращий спільний алгоритм шифрування зі списку клієнта. Він також надсилає свій цифровий сертифікат.

3) Authentication (Автентифікація). Клієнт перевіряє сертифікат сервера для підтвердження його справжності. Кожен такий сертифікат видається центром CA (Certification Authority) і містить відкритий ключ сервера.

4) Обмін ключами. Використовуючи відкритий ключ сервера, клієнт генерує та шифрує секретний ключ сеансу (session key) і надсилає його на сервер. Тільки сервер, маючи закритий ключ, може розшифрувати цей ключ.

5) Сеанс зашифрованого зв'язку. Після успішного обміну ключами обидві сторони надсилають повідомлення, що з'єднання готове. Від цього моменту весь трафік між ними шифрується за допомогою узгодженого симетричного ключа сеансу, який забезпечує високу швидкість передачі даних.

Протоколи безпеки SSL і TLS важливі в електронній комерції, веб-сервісах і месенджерах. Вони використовують як симетричне (AES), так і асиметричне (RSA, ECC) шифрування у різноманітних задачах, поєднуючи *швидкість симетричного* алгоритму шифрування AES та *зручність* управління ключами *асиметричного* шифрування. Взаємодія алгоритмів симетричного шифрування AES та асиметричного шифрування (RSA, ECC) у протоколах SSL та TLS реалізується наступним чином.

Перший етап. За допомогою асиметричного шифрування (RSA, ECC) надається захищений спосіб обміну спільним секретним ключом між клієнтом та сервером для подальшого шифрування протягом кожного сеансу. Це здійснюється тому, що передача симетричного ключа напряму через незахищений канал є вразливою. Асиметричне шифрування вирішує цю проблему, оскільки закритий ключ ніколи не передається. Клієнт використовує відкритий ключ від сервера для шифрування тимчасового ключа сеансу. Тільки сервер, володіючи своїм закритим ключем, може дешифрувати цей ключ. RSA широко застосовується для цифрових підписів, обміну ключами (наприклад, під час встановлення захищеного з'єднання TLS/SSL) та шифрування невеликих обсягів даних. Його недоліком є те, що він потребує ключ великого розміру (≥ 2048 біт) для надійного рівня безпеки, що уповільнює процес шифрування і дешифрування.

Другий етап. Після того, як за допомогою асиметричного шифрування був узгоджений симетричний «ключ сеансу», здійснюється симетричне шифрування AES. Клієнт і сервер використовують узгоджений ключ сеансу (наприклад, AES 256-біт), щоб зашифрувати та дешифрувати потоки даних протягом окремої сесії. Після завершення сесії цей ключ знищується. Такий спосіб забезпечує швидку та ефективну передачу великих обсягів даних. При цьому, симетричне шифрування (наприклад, AES) є значно швидшим і менш вимогливим до обчислювальних ресурсів, ніж асиметричне шифрування. Це критично важливо для безперебійної роботи веб-сайтів, відео-стримінгу та інших сервісів, де обробляються великі масиви інформації.

Об'єднання обох методів (симетричного та асиметричного) шифрування дозволяє отримати якісно новий результат, оскільки використання лише одного асиметричного шифрування для кожного пакету даних занадто повільне для передачі всього трафіку, а з іншого боку, симетричне шифрування вимагає попередньої домовленості про ключ, оскільки без асиметричного шифрування нема безпечного способу для обміну цим ключом між клієнтом і сервером.

З урахуванням вказаних вище пересторіг, а також характеристик алгоритмів асиметричного шифрування RSA та ECC, шифр ECC, виглядає більш адекватною ніж RSA альтернативою для багатьох застосувань у мобільних ad-hoc мережах Інтернету речей, роботизованих комплексах тощо, де швидкість і низьке енергоспоживання є критичними факторами.

Протоколи (стандарти) безпеки технології WiFi

Технологія WiFi на сьогодні є однією з найбільш поширених в мобільних ad-hoc мережах Інтернету речей. Її протоколи і стандарти безпеки змінювались в останні десятиліття від WEP-40 (Wired Equivalent Privacy, 1997, який досі використовується) до сучасного протоколу WPA3 (WiFi Protected Access 3, сертифікований з 2018 року), який зараз лише починає активно впроваджуватися (див. Таб. 1.2). Відповідно до структури таблиці 1.2, розглянемо ці протоколи у трьох аспектах: алгоритм шифрування (шифр, або сіпфер), спосіб автентифікації, крипто-протокол (спосіб побудови фрейму).

Протокол безпеки WEP. Стандарт IEEE 802.11 (1997, ратиф. 1999).

Алгоритм шифрування (шифр).

Сіпфер RC4 (потоковий по-символьний шифр з симетричним ключом розміром 1÷256 байт, див. с. 40). Фактично, WEP оперує тільки 2-ма типами ключів шифрування: 8 та 16 байт (64 біт та 128 біт).

Ключ 64 біт об'єднує спільний секретний статичний ключ 40 біт (пароль доступу що вводиться вручну, або автоматично надається точкою доступу, і є незмінним у всіх фреймах), а також відкритий динамічний ключ – вектор ініціалізації IV розміром 28 біт.

Вектор ініціалізації IV генерується програмним способом (псевдовипадково) для кожного фрейму. При достатньо великій кількості циклів роботи програмного генератора, псевдовипадковий вектор ініціалізації починає повторюватися (тобто, є періодичним).

Аналогічно, ключ 128-біт об'єднує секретний статичний ключ 104-біт та відкритий 24-бітний вектор ініціалізації IV.

Метод автентифікації та надання ключа

SKA (Shared Key Authentication). WEP-шифрування фрейму WiFi передбачає об'єднання попередньо визначеного статичного секретного 40/104-бітного ключа разом із 24-бітним динамічним відкритим ключем (вектором ініціалізації IV) у розширений 64/128-бітовий ключ , який генерується на передавальній стороні псевдо-випадковим чином для кожного чергового фрейму. Секретний ключ WEP може бути автоматично наданий точкою доступу WiFi для хоста локальної мережі WiFi шляхом довільного вибору з набору ключів за замовчуванням під час авторизації хоста. У точці доступу WiFi можна попередньо встановити до чотирьох ключів за замовчуванням; ключ за замовчуванням ідентифікується їх номером 0, 1, 2 або 3 у розширеному заголовку WEP-кадру. Секретний ключ WEP також може бути призначений як пароль доступу до локальної мережі WiFi з 5/13 символів у кодах ASCII або 10/26 символів шістнадцятирічних символів – HEX (40/104 біти).

Крипто-протокол (спосіб побудови фрейму).

«Крипто-протокол WEP» (не має окремої спеціальної назви).

Побудова фрейму WiFi (стандарт IEEE 802.11) на канальному рівні OSI згідно «крипто-протоколу WEP» представлено на рисунку 1.8, де ICV – “Integrity Check Value” (значення перевірки цілісності), IV – “Initiation Vector” (вектор ініціалізації). Структура фрейму включає заголовок (Header), корисне навантаження (Payload) та контрольну суму кадру (FCS).

На фізичному рівні доступу до середовища передачі електромагнітного сигналу (Media Access Layer – MAC), структура фрейму може мати версію 0 або 1 для відповідного протоколу (MAC 802.11). Натомість, загальна структура фрейму на канальному рівні є однаковою для обох цих версій [58].

Спосіб побудови фреймів WiFi за одним з перших протоколів безпеки («Крипто-протокол WEP») є достатньо типовим і має спільні риси з наступними протоколами безпеки WPA, WPA2, WPA3.

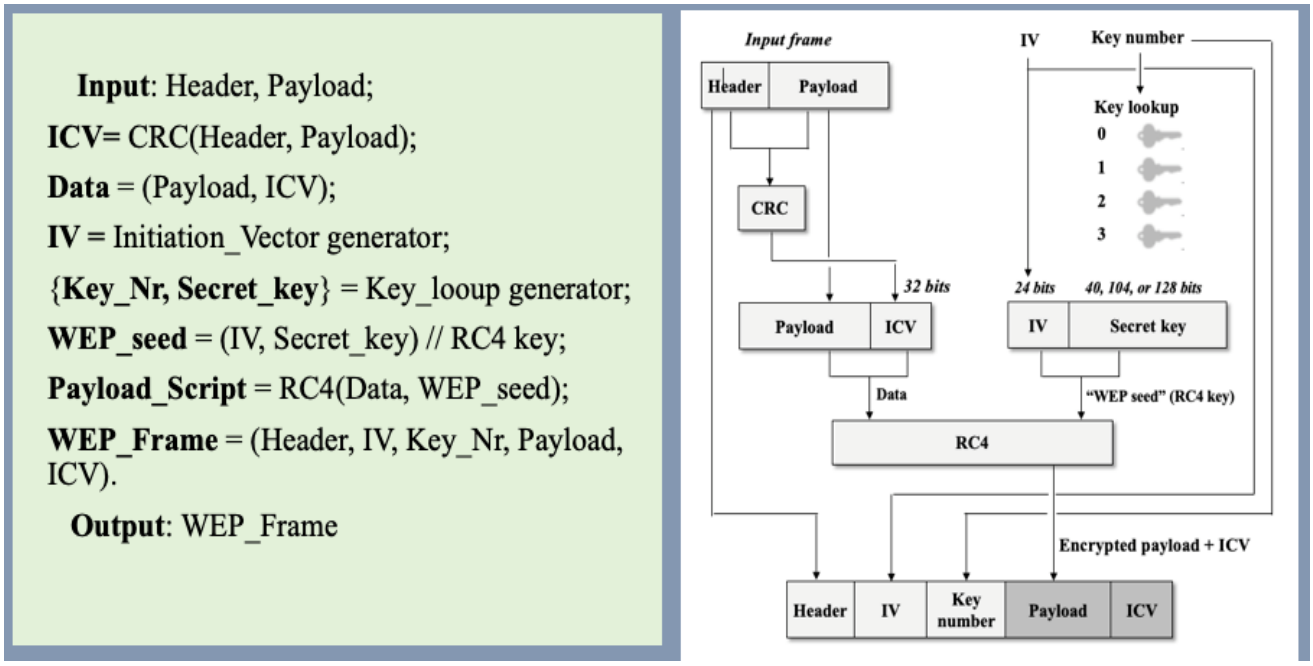


Рисунок 1.8 – Побудова фрейму WiFi 802.11 на канальному рівні OSI за «Крипто-протоколом WEP»

Протокол безпеки WPA (WiFi Protected Access, 802.11a/b/g/n/ac 2003).

Алгоритм шифрування (шифр). RC4 з трьома новими функцями:

- 64-бітну перевірку цілісності повідомлення MIC (Message Integrity Check) замість CRC (Cyclic Redundancy Check) для повторної ініціалізації порядкового номера щоразу, коли використовується новий часовий ключ;

- контроль послідовності пакетів для захисту від атак повторного відтворення (пакети, отримані не в порядку, відхиляються точкою доступу;

- у кожному черговому фреймі, попередньо узгоджений спільний секретний ключ (пароль доступу) доповнюється псевдо-випадковим вектором ініціалізації IV (Initialization Vector) в якості розширеного ключа шифрування WPA-PSK (WPA Pre-Shared Key), яким здійснюється для бездротової домашньої мережі WiFi чи малого офісу, де зазвичай немає централізованого сервера автентифікації типу RADIUS (Remote Authentication Dial-In User Service) [59].

Метод автентифікації (спосіб надання ключа). Два режими: а) WPA-Personal методом PSK ((Pre-Shared Key); б) WPA-Enterprise методом EAP (Extensible Authentication Protocol) з використанням сервера контроля доступу RADIUS (станд. 802.1X). Режим автентифікації WPA-Personal (PSK) оперує спільним паролем що вводиться у точці доступу і кожному клієнті. Автентифікація відбувається безпосередньо між кожним пристроєм і роутером шляхом «чотиристороннього рукостискання» (4-way handshake). Використовується для дома чи малого офісу. Режим автентифікації WPA-Enterprise методом EAP замість одного спільного пароля для всіх, передбачає окремі облікові дані для кожного клієнта (логін/пароль, сертифікат або токен). Роль сервера RADIUS: коли пристрій намагається підключитися до точки доступу WiFi, вона не перевіряє пароль сама, а пересилає запит на RADIUS-сервер (Authentication Server). Сервер звіряє дані у базі та каже «так» або «ні»..

Крипто-протокол (спосіб побудови фреймів): TKIP (Temporal Key Integrity Protocol). У 2002, “WiFi Alliance” затвердив перехідну версію протоколу WEP, відому як TKIP (Temporal Key Integrity Protocol) – під назвою WPA, (також її називають WPA1) [59].

Протокол безпеки WPA2 (стандарт IEEE 802.11i-2004).

Введений у 2001 році для мереж WiFi як наступник WPA зі стандартами IEEE 802.11a/b/g/n/ac. Став стандартом де-факто у 2004 році.

Алгоритм шифрування (шифр). Симетричний блочний шифр AES-128/256 (Advanced Encryption Standard). У побуті використовується AES-128 з ключом 128 біт. На відміну від RC4, AES вимагає спеціального апаратного прискорення, але при цьому забезпечує значно вищий рівень криптостійкості. Протокол безпеки WPA2 не просто використовує AES «як такий», натомість, він обгортає його в протокол CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol). Останній містить дві спеціальні функції: а) CTR (Counter mode) – використовується для шифрування даних, і перетворює блочний шифр AES на потоковий, що дозволяє шифрувати дані паралельно і швидко; б) CBC-MAC (інша назва CCMP) – Counter mode with cipher Block Chaining Message Authentication Code protocol для забезпечення цілісності повідомлення (гарантії того, що дані не були змінені зловмисником під час передачі). WPA2 підтримує зворотну сумісність зі старими пристроями та попередньою версією WPA з крипто-протоколом TKIP (Temporal Key Integrity Protocol) і сіпфером RC4, якщо у налаштуваннях роутера обрати "WPA2-TKIP"; при цьому швидкість обмежується до 54 Мбіт/с, а безпека падає до рівня WPA. На сьогодні, WPA2 вважається одним з найбільш захищених протоколів [59].

З 2023, AES-256 у режимі block-chain є шифрування за замовчуванням у додатках MS Purview Information., а за розрахунками фахівців, злам AES-256 на сучасних процесорах потребує мільйони років [60]. З точки зору сучасних вимог до захисту інформації в мобільних ad-hoc мережах IoT, протокол безпеки WPA2 має наступні вразливості. *Перше*. Механізм обміну ключами шифрування (4-way handshake) є недостатньо стійким до атаки KRACK (Key Reinstallation Attack); доведено, що ця атака може змусити клієнта перевстановити вже використаний ключ, що дозволяє дешифрувати трафік. *Друге*. Статичний алгоритм шифрування протягом всієї сесії уможливлює накопичення статистики взаємодії для перехоплення інформації.

WPA3. Стандарт протоколу шифрування WPA3 анонсовано у 2018 році альянсом WiFi як заміна WPA2. Він забезпечує кращий захист для окремих користувачів, більш надійну автентифікацію по паролю, навіть обраному користувачем не відповідно рекомендаціям щодо складності. Ця можливість реалізована завдяки механізму обміну ключами SAE (Simultaneous Authentication of Equals – стандарт IEEE 802.11s-2011/2012).

Метод шифрування стійкий до атак “offline dictionary”, коли зловмисник намагається визначити пароль, пробуючи можливі паролі без подальшої взаємодії з мережею. Використовує ключ 128-біт в режимі WPA3-Personal (192-бітове в WPA3-Enterprise). Стандарт WPA3 замінив ключ PSK протоколу WP2 на SAE згідно стандарту IEEE 802.11-2016, що призводить до безпечнішого початкового обміну ключами в особистому режимі. Альянс WiFi наголошує, що WPA3 мінімізує проблему слабких паролів і спрощує налаштування пристроїв. Версія протоколу WPA3-Enterprise пропонує розширюваний протокол автентифікації для захисту транспортного рівня OSI (EAP-TLS) через механізм обміну еліптичною кривою Діффі-Хеллмана (ECDH) та алгоритму цифрового підпису (ECDSA) з використанням 384-бітної еліптичної кривої [61].

Проведений огляд існуючих методів захисту інформації в каналах зв’язку IoT засвічує різноманіття загроз і методів протидії, а також зростаючі ризики кібератак, що потребує подальших теоретичних досліджень проблеми кіберзахисту, а також розробки нових методів підвищеного захисту інформації у розподілених комп’ютерних системах та мережах Інтернету речей з урахуванням виявлених недоліків та особливостей мережевих інтерфейсів на різних рівнях взаємодії.

Перспективним напрямком в теорії кодування і захисту інформації в системах та мережах IoT є представлення цифрових потоків послідовністю команд у ієрархічних формальних граматиках, рівні якої відповідають інтерфейсам обраної моделі взаємодії відкритих систем в мережі IoT. Це дозволяє розділити складну задачу наскрізного кодування на окремі складові частини.

1.3 Моделі і методи управління потоками даних в мережах

Далі стисло оглядаються існуючі традиційні та перспективні моделі і методи управління потоками даних в телекомунікаційних мережах у контексті їх можливого застосування для оптимізації розподілу даних в комп'ютерних системах та мережах Інтернету речей (IoT).

Одним із способів збільшити стійкість мережі до впливу негативних факторів є оптимізація цифрових потоків для найкращого використання наявних ресурсів у змінних ситуаціях (виходу із ладу її елементів, зовнішнього втручання, тощо). Важливим критерієм використання наявних ресурсів є сумарний інформаційний потік мережі, а завдання його підвищення відомо як проблема максимального потоку мережі – "MaxFlow problem" (MFP).

Історично, проблема MFP виникла в логістичних мережах для доставки деякого однорідного продукту (газу, нафти тощо) від місця виробника S до терміналу накопичування T. Класична постановка проблеми MFP наступна.

1) Логістична мережа та її транспортна система оперують потоками одного типу (одно-продуктовими).

2) Потоки не спрямовуються в протилежних напрямках окремих ланок та логістичної мережі в цілому, а тільки від джерела S до терміналу T.

3) Канали і потоки від джерела S до термінала T не перетинаються.

4) В логістичній мережі відсутні проміжні джерела та накопичувачі потоків, окрім джерела S (виток) і терміналу накопичення продукту T (стік).

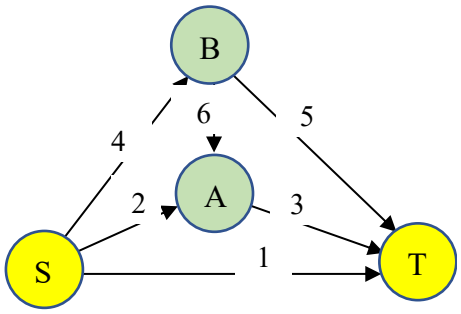
5) Моделлю логістичної мережі є *замкнутий строго-орієнтований двополюсний планарний (плоский) зважений* ST-граф G(V) з двома особливими вершинами S і T (полюсами графу), де односпрямовані зважені дуги визначають транспортні канали певної пропускної спроможності, а візуальна форма графу відображується на площині без перетину дуг [62].

Логістичний підхід до проблеми MFP певною мірою адаптований для оптимізації інформаційних потоків, натомість, має свої обмеження і недоліки через недостатню адекватність сучасним комп'ютерним системам і мережам Інтернету речей з динамічно конфігурованими каналам зв'язку [63].

Розглянемо властивості планарного графу як логістичної моделі мережі в задачах оптимізації потоків на прикладі замкнутого зваженого планарного ST-графу G(V) з чотирма вершинами (V=4), представленого на рисунку 1.9 у трьох еквівалентних формах: візуальній, матричній, алгебраїчній [63].

Два *полюси* графу (вершини S і T) не мають зовнішніх дуг, а вершини (A і B) є внутрішніми, тобто граф є *замкнутим*. Кожна впорядкована пара вершин (V_K, V_N) визначає окрему дугу спрямовану від V_K до V_N. Кожна дуга має «вагу» C – пропускну спроможність (Capacity) в напрямку від V_K до V_N. Будемо позначати ці дуги трійками (V_K, V_N, C), в яких порядок запису вершин має значення. Скажімо, у графі на рис. 1.9 є дуга (S, B, 4), натомість немає дуги (B, S, 4); тобто, формально – це «нуль-дуга» (B, S, 0). Для зручності пояснень, ваги дуг обрано цілими числами від 1 до 6 по кількості дуг на графі, що дозволяє ідентифікувати кожну окрему дугу своєю вагою.

Максимально можлива кількість дуг на довільному строго-орієнтованому графі з кількістю вершин V (не обов'язково планарному), очевидно, становить: $N_{дуг}(V) = (V^2 - V)/2 = V \cdot (V-1)/2$. В нашому випадку, маємо $N_{дуг}(4) = 4 \cdot (4-1)/2 = 6$, тобто ST-граф G(V) на рис.1.9 є *повним* планарним строго-орієнтованим графом (з максимально можливою кількістю одно-спрямованих дуг).



а) Візуальна форма графу

	S	A	B	T
S		2	4	1
A	0		6	3
B	0	6		5
T	1	0	0	

б) Матрична форма графу

{(V_K, V_N, C)} = {(S, T, 1), (S, A, 2), (A, T, 3), (S, B, 4), (B, T, 5), (B, A, 6)}

в) Алгебраїчна форма графу (список дуг)

Рисунок 1.9 – Форми ST-графу G(V=4): а) візуальна; б) матрична; в) алгебраїчна

Довільний ST-граф G(V) з кількістю вершин V≤4 завжди є планарним, тобто, його можна відобразити на 2D-площині безе перетину дуг, і при цьому він може бути повним, тобто мати максимальну кількість дуг $N_{дуг}(V)\leq 6$. Натомість, ST-графи з кількістю вершин V≥ 5 мають певні обмеження по кількості дуг, інакше вони не будуть планарними, тобто, їх неможливо буде відобразити у 2D-площині без перетину хоча б однієї пари дуг.

Наприклад, ST-граф G(5) має максимально можливу кількість дуг 9, що на 1 дугу менше, порівняно з вказаною вище формулою $N_{дуг}(5) = 5\cdot(5-1)/2 = 10$. Граф G(6) має максимальну кількість дуг 12, тобто, на 3 дуги менше, порівняно з $N_{дуг}(6) = 6\cdot(6-1)/2 = 15$. В загальному випадку, максимальна кількість дуг планарного ST-графу G(V) обчислюється за комбінаторною формулою [63]:

$$PGE\ (V) = 3(V-2),\ V \geq 3. \qquad (1)$$

Це означає, що планарні графи G(V) з кількістю вершин V≥5 є неповними відносно графів загального типу (не обов'язково планарних), натомість, планарний граф G(V), що має максимально можливу кількість дуг згідно формули (1), є «*повним планарним графом*» серед множини інших можливих (планарних) ST-графів [63].

Розглянемо існуючи підходи до проблеми MFP про потоки мережі.

Відомим рішенням проблеми MFP про максимальний потік на ST-графі G як моделі логістичної мережі є алгоритм («метод») Форда-Фалкерсона (позначимо його FFA). На кожній ітерації «K», FFA здійснює пошук чергового можливого шляху P_K в напрямку від S до T ("augmenting ST-path") на остаточному графі (residual graph G_R) способом «пошук в глибину» (depth search first, DFS), обчислює доступний на шляху P_K додатковий частковий потік f_K (feasible partial flow) в напрямку від полюса S до полюсу T, після чого зменшує вагу кожної дуги на шляху P_K на величину f_K. Алгоритм закінчує свою роботи, коли в остаточному графі G_R не залишиться жодного можливого шляху P_K для додаткового потоку f_K. Сума усіх часткових потоків $sum\{f_K\}$ приймається за кінцевий результат – максимально можливий потік на графі G ([64], 1956).

.

Алгоритм Форда-Фалкерсона (FFA) має псевдо-поліноміальну обчислювальну складність реалізації O(F·E), де O – загально прийнята нотація функції складності по кількості необхідний ітерацій алгоритму (елементарних базових операцій) або часу на його виконання; F – максимальний ST-потік, E – кількість дуг у ST-графі. Наприклад, для графу G на рис. 9 (F=7, E=6) алгоритм FFA дає оцінку складності O(F·E)=(7·6)=42. За цієї оцінки, алгоритм FFA краще використовувати для невеликих значенням максимального потоку.

На основі методу FFA були розроблені альтернативні підходи до проблеми максимального потоку на графі MFP. Одним з найбільш поширених і популярних є алгоритм блокуючого потоку Дініца. На відміну від FFA, він реалізує пошук ST-шляху на графі G(V) способом «в ширину» (breadth first search, BFS), і має обчислювальну складність $O(V^2 \cdot E)$, де V - кількість вершин графу G; E - кількість дуг ([65], 1970). Для графу G на рисунку 9 алгоритм Дініца має оцінку складності $O(V^2 \cdot E)=(4^2 \cdot 6) = 96$. За такої оцінки, алгоритм Дініца краще використовувати для графів з малою кількістю вершин.

Подібно до алгоритму Дініца (з деякими відмінностями), метод FFA реалізовано в алгоритмі Едмондса-Карпа [66], 1972) що має оцінку складності $O(V \cdot E^2)$. Так, для графу G (рис. 9) алгоритм Едмондса-Карпа дає оцінку складності $O(V \cdot E^2) = O(4 \cdot 6^2) = 144$. Для певної топології і метрики графу, алгоритм Едмондса-Карпа може перевершити алгоритми FFA і Дініца за швидкодією. Зокрема, його доцільно застосовувати для графів з великою кількістю вершин та малою кількістю дуг (т. зв., *розріджені* графи).

Для ST-графу, в якому вага дуг може приймати тільки два можливих значення (0 чи 1), ефективним є алгоритм (Goldberg-Tarjan) що імітує фізичний принцип розподілу потоків води в мережі трубопроводів під високим тиском – т. зв. push-relabel метод («проштовхування–пере-маркування»). Він припускає «надлишковий потік» (excess preflow) на вході вершин графа (окрім S). Цей алгоритм, з використанням методу «блокуючого потоку» Дініца, має оцінку складності $O[V \cdot E \cdot \log_2(V^2/E)]$ ([67], 1986).

Для деяких окремих випадків структури ST-графу (бінарні ваги робер), альтернативний алгоритм Голдберга-Рао може перевищити швидкодію алгоритму Голдберга-Тар'яна, і демонструє оцінку складності O(V·E), де V, E - кількість вершин і дуг на графі відповідно ([68], 1998).

У 2006 році Д. Папп (D. Papp) представив удосконалений алгоритм Голдберга-Рао для розв’язання задачі максимального потоку MFP. Він працює не лише для графів з бінарною ємністю дуг, але й для більш загального випадку (вага дуги може бути цілим, або невід’ємним дійсним числом) [69].

Оригінальний алгоритм для *приблизної оцінки* максимального потоку за майже лінійний час запропоновано П. Крістіано (P. Christiano) та ін. для зваженого *неорієнтованого* ST-графа. Він використовує відомий метод матриць Лапласа для розв'язання системи лінійних рівнянь для електричного потоку. У роботі стверджується, що даний алгоритм є найшвидшим серед відомих на той час для розв’язання задачі MFP у першому наближенні ([70], 2010).

Формалізований підхід до розв’язання задач комбінаторної оптимізації, таких як проблема максимального потоку MFP, можна побудувати на основі принципу максимуму шляхом вибору деяких необхідних умов екстремуму пошука рішення (наприклад, способу обрання шляхів мережі для потоків, а також критерія призначення потоків на цих шляхах), які мають дотримуватися на кожному кроці ітераційного алгоритму. Це дозволяє заздалегідь відсіяти неприпустимі комбінаторні варіанти пошуку оптимального рішення задачі, і тим самим, скоротити кількість необхідних ітерацій та прискорити алгоритм за рахунок відбору компактної множини перспективних кандидатів на кожному кроці алгоритму. За певних обставин, сукупність необхідних умов екстремуму, може виявитися (теоретично чи емпірично) також і достатньою для оптимальності алгоритму с точки зору загального критерія, наприклад, максимального сумарного потоку мережі ([71], 2013).

Теоретичні основи розв’язання задачі про максимальний потік (MaxFlow problem MFP) в загальному контексті використання різних типів графу як моделі мережі, викладено в [71] (1976).

Змістовний огляд методів і алгоритмів для обчислення максимального потоку на замкнутому орієнтованому зваженому ST-графі відображено у книзі Д. Вільямсона (D. Williamson). Книга розширює рамки дослідження мережевого потоку для задач з кількома товарами та надає детальний аналіз конкретного випадку з двома товарами. Також визначено поняття ST-потоку в неорієнтованому графі, де будь-яке ребро має довільну позитивну орієнтацію та складається з двох протилежно спрямованих дуг однакової ємності ([72] (2019).

Розширений підхід до розв'язання задачі MFP і моделювання інформаційних потоків в програмно-конфігурованих мережах представлено в роботі [73] (2019). На відміну від традиційної логістичної моделі у вигляді біполярного замкнутого ST-орієнтованого зваженого планарного графу (ST-DWPG), пропускна здатність (максимальний потік) мережі визначається на відкритому 3-полюсному (не обов'язково планарному) вільно орієнтованому графі. Відповідний алгоритм для 6-вершинного графа представлено в [74] (2019).

Важливий клас задач з оптимізації потоків електроенергії – (Equal Maximum Flow Problem - EMFP) досліджено в [75] (2020). Метод EMFP прагне досягти максимально рівного потоку на всіх ребрах у підмножині ребер графа. Також розширено задачу на «майже рівний максимальний потік» (Almost Equal Maximum Flow Problem - AEMFP), де значення електричного потоку в окремих ланках мережі відрізняються від середнього «майже рівного максимального потоку» відповідно до певної функції відхилення. Фізичні закони розподілу потоків електроенергії в електричних схемах мають деякі схожі властивості з інформаційними потоками в телекомунікаціях, і тому представляють певний інтерес у загальному контексті проблеми MFP.

У роботі [76] (2021) запропоновано майже-оптимальний алгоритм маршрутизації (near optimal routing algorithm) в мережі SDN, заснований на теоремі про максимальний потік/мінімальний розріз. Показано, що розподіл потоку маршрутах мінімального транзиту підвищує продуктивність мережі порівняно з традиційними алгоритмами пошуку найкоротшого шляху.

Для оптимізації різнотипних (багато-продуктових, або мультимодальних) потоків у логістичних системах, відомо про застосування т. зв. генетичного алгоритму що зазвичай використовується в задачах маршрутизації транспортних засобів (Vehicle Routing Problem - VRP). Він імітує природну еволюцію Дарвіна в моделі транспортної мережі, та включає п'ять кроків переробки: кодування завдання, ініціалізація популяції, визначення селективних операцій, мутація, процес відбору ([77], 2022).

В статті [78] (2023) представлено огляд детермінованих задач (в яких усі вхідні дані і параметри, обмеження та цільова функція відомі наперед і не є випадковими чи невизначеними) логістичної оптимізації для знаходження максимально можливого потоку в мережі (при мінімальній вартості витрат) на моделі орієнтованого зваженого ST-графа, з акцентом на найгірший випадок відносно тривалості часу на розв’язання задачі.

Стаття відображує лише ті алгоритми, що мають швидший час виконання, ніж опубліковані раніше. Зазначено, що алгоритм Голдберга-Рао має слабо-поліноміальний час виконання, а існування сильно-поліноміального алгоритму з оцінкою O(V·E) було показано Орліном у 2013 році, а також Орліном та Гонгом у 2021 році. Крім того, розглянуто методи внутрішніх точок (Interior Point Methods - IPM) для MFP-рішень, а також алгоритми мінімізації вартості загального потоку мережі ([78]).

Перспективним напрямком у вирішенні складних комбінаторних задач, таких як проблема про максимальний потік мережі (MaxFlow problem - MFP), є використання технологій штучного інтелекту (Artificial Intelligence - AI) та машинного навчання (Machine Learning ML). Огляд сучасних рішень у цій сфері опубліковано в [79] (2021). Через складний характер цих задач, розглянуті алгоритми, значною мірою, покладаються на вручну-розроблені евристичні правила прийняття рішень, які в іншому випадку є занадто дорогими для обчислення або математично недостатньо чітко визначеними. Таким чином, машинне навчання виглядає природним кандидатом для розв’язання подібних задач більш формалізованими та оптимальними способами.

В роботі [80] (2021) описано метод планування для комп’ютера з ресурсно-залежним часом обробки даних і прийняття рішень. Представлено евристичні алгоритми для розв'язання складних задач (з не поліноміальною обчислювальною складністю – т. зв. NP-повних проблем) на спрощених та уніфікованих моделях. Визначено три типи завдань планування: терміни призначення завдань, розподіл ресурсів між операціями завдань та планування завдань машин. Методи глибокого навчання (Deep Learning - DL) на основі графів розглянуто в [81] (2022). Основну увагу приділено орієнтованим та неорієнтованим (дротовим та бездротовим) програмно-конфігурованим мережам SDN як перспективному рішенню, що спонукає мережеву індустрію до перегляду традиційної мережевої архітектури. Для відстеження подальших інновацій у напрямку глибокого машинного навчання, створено публічний репозиторій GitHub з постійно обновлюваними статтями.

Оригінальний метод вирішення багато-функціональних задач шляхом машинного навчання представлено в роботі Н. Байчика (N. Baycik), яка поєднує «дерево рішень» (decision tree) та «регресію випадкового лісу» RFR (Random Forest Regression) – алгоритм навчання з учителем через ансамбль з незалежних дерев рішень (ft. зв. forest - ліс). Дерево рішень та ансамбль (ліс) рішень будуються незалежно одне від одного ([82], 2022).

У роботі [83] (2023) запропоновано інтелектуальний спосіб маркування вузлів (Intelligent Node Labelling - INL) для розв'язання задачі максимального потоку на орієнтованому ST-графі мережі. Він оминає поширений спосіб «доповнюючих шляхів» (“augmenting paths”) притаманний відомим підходам до обчислення максимального потоку на основі алгоритму Форда-Фалкерсона (FFA). Натомість, вхідний та вихідний потік між полюсами S і T збалансовується для проміжних вузлів для усунення надлишкового або застійного потоку, зменшуючи недостатньо використані дуги відтоку. Алгоритм потребує щонайбільше дві попередні операції для перетворення початкової мережі з V вузлів на еквівалентну мережу зі складністю O(V·E) у найгіршому випадку.

Підсумовуючи методи розв’язання проблеми MFP (MaxFlow Problem), відомої в логістичних системах ще з середини минулого століття, зауважимо, що розрахунок максимально можливого потоку між двома «полюсами» мережі (джерелом витоку S та терміналом накопичення T) досі залишається актуальною науково-прикладною задачею, зокрема, для сучасних мобільних ad-hoc мереж, в яких є два спеціальні об’єкти оперативного управління і зв’язку з базовою станцією. Про це свідчать наведені вище публікації останніх років, дотичні до MFP, основним напрямком досліджень і метою яких є зменшення алгоритмічної складності і часу на прийняття оптимальних рішень.

Незважаючи на вагомі наукові результати і практичні рішення в даному напрямку, існуючі підходи до розв’язання проблеми MFP *ще не вичерпали можливості підвищення швидкодії відповідних алгоритмів*, і окрім того, мають *інші прогалини*, критичні с точки зору оперативного управління мобільними ad-hoc мережами Інтернету речей в екстремальних умовах реального часу при жорстких обмеження до апаратно-програмних і енергетичних ресурсів.

Перше. Через застарілу логістичну модель мережі (орієнтований зважений ST-граф), відомі рішення не враховують властивість сучасних програмно конфігурованих мобільних каналів зв’язку, внаслідок чого, не підтримують можливість збільшення максимального потоку мережі за рахунок динамічної реконфігурації пропускної спроможності каналів зв’язку при швидких випадкових змінах структури мережі (втрата інтерфейсів чи самих об’єктів мобільної ad-hoc мережі, зміна їх взаємного положення у просторі, тимчасові чи ситуаційні порушення зв’язку між ними тощо). Застосування неорієнтованих графів для моделювання та розрахунку потоків розуміється як симетрично орієнтований граф, що не відображує реконфігурацію каналів.

Друге. Обчислення самої величини максимально можливого потоку між двома виділеними об’єктами мережі (джерелом S» і терміналом накопичення T) недостатньо для ефективного управління мобільною ad-hocмережею Інтернету речей. Треба ще розрахувати конкретний розподіл потоків даних по окремих об’єктах і каналах зв’язку мережі.

Висновки до розділу 1

В даному розділі дисертаційної роботи проведено аналітичний огляд проблеми захисту інформації і управління розподілом даних в комп'ютерних системах та мережах у зв'язку із стрімким впровадженням мобільних мереж Інтернету речей (IoT) у промисловість, оборону і повсякденне життя. На основі системного підходу, досліджено розвиток еталонних моделей Інтернету у контексті інформаційної безпеки та сучасні криптографічні технології Інтернету речей. Проаналізовано існуючі протоколи і стандарти захисту в безпровідних інтерфейсах WiFi, а також моделі і методи управління цифровими потоками.

Аналіз робіт у сфері захисту інформації і управління розподілом даних в комп'ютерних системах та мережах показав, що на поточний момент створено ефективні еталонні безпекові моделі Інтернету речей від організацій IoT-WF, NIST. Досягнуто високої криптографічної стійкості шифрів і крипто-протоколів (шифр AES, протокол GCMP, автентифікація SAE, стандарт WPA3). Існують швидкі моделі та алгоритми оптимізації потоків даних в мережах ("Push-Relabel"). Потужним інструментом захисту і оптимізації мереж IoT є використання ШІ. Разом з тим, існують невирішені питання та виклики, зокрема, консерватизм застарілих технологій, що стримує втілення нових безпекових рішень поверх наявних, а також реактивний підхід до захисту даних в комп'ютерних мережах (виявлення загроз по факту). Окрім того, є недостатньою швидкодія оптимізації потоків реального часу в мобільних ad-hoc мережах. Залишається можливість додаткового збільшення максимального потоку шляхом динамічної ре-конфігурації каналів зв'язку при швидких змінах структури мережі. Зроблено висновок, що існуючі рішення недостатні для комплексної підтримки сучасних вимог до кодування, захисту і управління даними в мобільних ad-hoc мережах IoT. Є потреба у пост-криптографічному захисті даних незалежно від існуючої апаратної підтримки, проактивних методах динамічного кодування захисту і розподілу даних, усунення вразливості застарілих операційних технологій, а також зменшення впливу людського фактору у питаннях кібербезпеки.

У зв’язку з цим, поставлено завдання щодо створення методу крипто-стійкого динамічного кодування на основі багаторівневої формальної граматики і побудови недетермінованої прикладної машини Тьюринга, а також комп’ютерних алгоритмів крипто-стійкого динамічного кодування і розподілу даних ad-hoc мережі Інтернету речей.

РОЗДІЛ 2
СИНТЕЗ МЕТОДУ КРИПТО-СТІЙКОГО ДИНАМІЧНОГО КОДУВАННЯ ДАНИХ AD-HOC МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ

В даному розділі дисертаційної роботи обґрунтовуються основні теоретичні положення щодо синтезу моделі і методу крипто-стійкого динамічного кодування даних для підвищення рівня кіберзахисту мобільної ad-hoc мережі Інтернету речей.

Удосконалюється модель динамічного кодування даних мобільної ad-hoc мережі IoT на основі багаторівневої формальної граматики. Досліджується еволюція екосистеми робототехніки “Robotics Eco-System”, методи і моделі автоматизованого проектування роботизованих мобільних комплексів і платформ.

Аналізуються особливості кодування і структуризації даних у технологіях локальних мереж Ethernet та її бездротового аналогу технології WiFi. Обґрунтовується багаторівнева формальна граматика для гнучкої структуризації фреймів мобільної ad-hoc мережі Інтернету речей.

Розглядається узагальнена функціональна схема дистанційного управління мобільною платформою Інтернету речей, а також трирівнева структура роботизованого мобільного комплексу Інтернету речей. Пропонується удосконалена модель динамічного кодування даних мобільної ad-hoc мережі Інтернету речей на основі багаторівневої формальної граматики.

Розробляється прикладна машина Тьюринга для гнучкого кодування фреймів в каналі зв’язку. Визначається система команд і узагальнена формальна граматика прикладної машини Тьюринга з правилами семантики потокового кодування фреймів. Будується недетермінована прикладна машина Тьюринга для динамічного кодування даних в каналі зв’язку Інтернету речей. Синтезується метод крипто-стійкого динамічного кодування даних ad-hoc мережі IoT на основі недетермінованої машини Тьюринга.

2.1. Удосконалення моделі динамічного кодування даних мобільної ad-hoc мережі IoT на основі багаторівневої формальної граматики

У даному підрозділі досліджується еволюція екосистеми робототехніки “Robotics Eco-System” (RES), методи і моделі автоматизованого проектування роботизованих мобільних комплексів і платформ. Аналізуються особливості кодування і структуризації даних у технологіях локальних мереж Ethernet та її бездротового аналогу технології WiFi. Обґрунтовується багаторівнева формальна граматика для гнучкої структуризації фреймів мобільної ad-hoc мережі Інтернету речей. Розглядається узагальнена функціональна схема дистанційного управління мобільною платформою Інтернету речей, а також трирівнева структура роботизованого мобільного комплексу Інтернету речей “Robotics Interoperability Framework” (RIF). Пропонується удосконалена модель динамічного кодування даних мобільної ad-hoc мережі Інтернету речей на основі багаторівневої формальної граматики.

Еволюція технологій і моделей екосистеми робототехніки RES

Сучасні мобільні роботи використовують автономні інтелектуальні транспортні засоби (Autonomous Intelligent Vehicles - AIV) з динамічним керуванням рухом, навігацією та функціями когнітивного прийняття рішень, що базуються на бездротових каналах зв'язку, аналізі даних датчиків у режимі реального часу та методах штучного інтелекту (AI).

Типовий безпілотний літальний робот (дрон) зазвичай виконує заздалегідь запрограмоване завдання з втручанням людини або без нього. Особливий інтерес становлять «соціально інтерактивні роботи», для яких важлива взаємодія з людиною [84].

У 2006, було представлено одного з перших польових роботів з автономною навігацією між рядами кукурудзи і трьома варіантами взаємодії з ноутбуком оператора (Personal Digital Assistant, PDA): пряме Ad-Hoc-з`єднання; з'єднання на основі точки доступу WiFi; з'єднання через Інтернет [85].

.

У 2008, H. Haehnel представив літаючого робота з дистанційним керуванням на відкритій веб-платформі MoveIT (розробки робото-технічних додатків) для управління складними системами через порівняльний аналіз поточних результатів з найкращими «прототипами» (“benchmarking”) [86].

У 2009, R. Laird розглянув стан розвитку безпілотних систем у галузі оборонних закупівель дійшовши висновку, що державна політика має передбачати заміну пілотованих систем на безпілотні роботизовані пристрої IoT, а також використання інтелектуальних роботів в якості помічників для військово-службовців чоловічої та жіночої статі [87].

У 2012, представлено роботизовану платформу з управлінням через Інтернет в реальному часі і певним рівнем автономності завдяки вбудованій системі навігації. Апаратна реалізація платформи містить мікроконтролер NXT Intelligent Brick, цифрову камеру, інтерфейси WiFi/3G/GPS, і використовує технологію LEGO® для швидкої розробки робото-технічних прототипів. Платформа масштабується для деяких інших завдань. Але у промисловості керування по каналах Інтернет є проблематичним через високу затримку та низьку стабільність IP-з’єднання [88].

У 2013, професор P. Springer (США) оприлюднив військові інновації та вразливостей в асиметричних війнах з бойовими роботами, підкреслив роль AI і теорії автоматів Тюрінга. Навів перелік центрів AI та робототехніки (National Defence Authorization, Air Force Research Lab, DARPA, Robotic Systems Joint Project Office, Boston Dynamics, Carnegie Mellon University Robotics Inst, Foster-Miller Co, General Atomics, MIT Computer Science and Artificial Intelligence Lab, Stanford AI Lab, BAE Systems, Boeing, European Aeronautic Defence and Space Co, Lockheed Martin Corp, Northrop Grumman Corp, QinetiQ North America, Raytheon Co), а також пов’язані інформаційні ресурси і дослідники [89].

У 2014, індійські інженери показали безпілотну платформу з камерою для контроля кордонів і бойових дій через бездротовий канал ZigBee/Інтернет (прототип TALON Foster-Miller), що пересувається по піску, снігу та воді (глибиною до 100 футів), а також підніматися сходами [90].

У 2015, представлено робот з необмеженим радіусом керування, контрольований через смартфон двох-тональним багато-частотним декодером HT9170 (Dual Tone Multi-Frequency, DTMF). Обробку даних здійснює бортовий МК LPC2148, а керування – двигуни постійного струму. Мікроконтролер налаштовано в середовищі проектування “Keil μVision-4 IDE” з графічним інтерфейсом “Flash Magic” (програмна утиліта для прошивки МК). Оператор може рухати робота вліво/вправо-вперед/назад та зупиняти [91].

Того ж 2015 року, турецькі інженери розробили мобільний робот для небезпечних умов (платформа з рукою- маніпулятором, 4 ступені свободи із захопленням) з управлінням людиною через ПК або контролер. Функції робота «сприйняття вдень/вночі» забезпечується бортовими камерами та лазерним світлодіодом. Радіозв’язок з оператором реалізується в діапазоні 433 МГц трьома 15-канальними модулями [92].

У 2016, в’єтнамські вчені представили робота, керованого оператором через ПК з графічним інтерфейсом та мобільний Інтернет 3G [93].

У 2017, турецькі інженери представили прототип мобільної платформи-носія з фіксованою камерою переднього огляду, а також бойової платформи з двома стерео-камерами на вбудованій «руці-маніпуляторі». Відеодані з камер передаються через протокол UDP до локального IP-оператора. Передня відеокамера дозволяє відстежувати робота, а стерео на VR-окулярах (virtual reality glasses) дає реалістичне бачення навколишнього середовища [94].

Обертання платформи зброї, отримане з інерціального вимірювального блоку (IMU), адресується клієнту. Кутові рухи джойстика оператора надсилаються роботу: тангаж (pitch) – обертання об'єкта навколо поперечної (бічної) осі Y, наприклад, підняття або опускання передньої частини (носа) об’єкта; рискання (yaw) – обертання об’єкта вправо-вліво навколо вертикальної осі Z, залишаючись в одній площині руху та незмінній висоті. Бойова платформа є інваріантною до руху платформи-носія та фіксується на цілі за допомогою лазерного вказівника. Прототип зосереджується переважно на динаміці керуванні роботом [94].

У 2018/2023, на веб-порталі IEEE оприлюднено інтерактивний посібник з робототехніки, який отримав нагороду IEEE (E. Guizzo – директор з цифрових інновацій в IEEE “Spectrum” та співзасновник IEEE “Robots Guide”) [95]. У березні 2019, Toyota Motor розробила робота (Human Support Robot - HSR) як дослідницьку платформу побутового мобільного маніпулятора для фізичної роботи і комунікації, де розглянуто кінематики робота [96]. У квітні 2019, вчені з Політехнічного університету Валенсії (Іспанія) опублікували огляд мобільних роботизованих систем (189 джерел), де відображено питання класифікації, пересування, сприйняття, пізнання, навігації тощо, а також нові тренди і [97].

У 2020, А.М. Романов (РТУ) надав огляд апаратно-програмного керування роботами в небезпечних умовах (космос, мілітарі), та алгоритм управління з фільтром Калмана на FPGA і дистанційною ре-конфігурацією. Для розрахунку регуляторів в реальному часі розроблено процесор (типу “Leon 3”) для трудомістких операцій (навчання ШІ). Для внутрішнього зв’язку використовуються Ethernet і протокол “Space Wire” (до 400 Мбіт/с на відстані 10 м). Окреслено проблеми підводної робототехніки (навігація, зв'язок та герметизація маніпуляторів), а також стратегічних/тактичних контролерів на базі вбудованих комп'ютерів з OS Linux/“QNX OS”. Зазначено, що галузь робототехніки має різноманіття інтерфейсів і компонентів: шина CAN (Controller Area Network) – стандарт обміну повідомленнями між електронними блоками керування (ECU) усередині автомобілів, а також для використання в промисловій автоматизації та вбудованих системах; технологія Ethernet, послідовний інтерфейс RS-232, операційна система робота (ROS), аматорського радіо-пристрої та програмне забезпечення ArduPilot [98].

ArduPilot – це популярний відкритий пакет, дотичний до контролера Arduino, для управління БПЛА (наземні, підводні тощо). Бібліотека ArduPilot має набір «прошивок» для ArduCopter (мультикоптери), ArduPlane (літаки з нерухомим крилом), ArduRover для поверхневих (наземних/надводних) роботів та човнів; ArduSub для підводних апаратів типу ROV (Remotely Operated Vehicle) та AUV [99].

Платформа ROV – це підводний неавтономний дистанційно керований по кабелю мобільний об’єкт (т. зв. умбілікус, або кабель-пуповина). Платформа AUV (Autonomous Underwater Vehicle) – це підводний апарат, який працює без прямого керування людиною і не має кабелю. Він працює за попередньо запрограмованою місією, використовує власний бортовий комп’ютер і навігаційні системи (інерційну, акустичну тощо) для самостійного виконання завдання. Після запуску він сам визначає своє місцезнаходження, уникає перешкод та збирає дані. Після завершення місії повертається на поверхню або до базового судна. Призначені для океанографічних досліджень, пошуку об’єктів на великій території та військових цілей) [100].

У листопаді 2020, L. Tsiu та E. Markus (Central Univ. of Technology, Free State prov. of South Africa) опублікували дослідження щодо методів управління командою (роєм) мобільних роботів, які мають широке застосування (пошук та рятування, безпека, спостереження), оскільки вони пропонують більшу ефективність, гнучкість, резервування та маневреність. Кожен робот у складі рою збирає інформацію про своє середовище та спілкується із сусідами, щоб прийняти власне рішення, тим самим зменшуючи трафік зв'язку та час обробки. Централізоване управління є більш точним та надійним, але коштує дорого. Децентралізований підхід до управління набув популярності з роками. Деякі дослідники мають альтернативні підходи до управління роєм роботів [101].

Нещодавно, австралійські вчені M. Moniruzzaman та ін. опублікували огляд методів дистанційного управління мобільними роботами для автоматичного виявлення підводних об’єктів, а також власні дослідження щодо керування роботизованими транспортними засобами. Зазначалося, що у випадку складних завдань управління роботами в реальному часі (таких як безпілотні літальні апарати, безпілотні літальні апарати та дистанційні підводні апарати) необхідне вдосконалення традиційних підходів, оскільки методи дистанційного контролю (особливо двосторонні) для стаціонарних маніпуляторів та мобільних роботів суттєво відрізняються [102].

У квітні 2022, італійські дослідники E. Maset та ін. представили експериментальне дослідження продуктивності мобільного робота з лазерним сканером та інерціальним вимірювальним блоком для 3D-топографічного картографування, яке продемонструє доцільність і точність автоматичного збору даних відносно ручних топографічних операцій [103].

У жовтні 2022, нігерійські інженери продемонстрували простого мобільного робота на мікрокомп'ютері Raspberry-Pi3 для освітніх та дослідницьких цілей, який має вбудовану аудіо/відео камеру, дистанційне керування через інтерфейс WiFi смартфона з ОС Android та інтерфейсом GUI (Graphical User Interface). Відзняті матеріали зберігаються в пам'яті мобільного пристрою. Робот використовує локальний алгоритм навігації з УЗ-датчиками. Дистанційне програмування та управління бортовим МК Raspberry використовує протокол SSH (Secure Shell), призначений для безпечної передачі даних через незахищений або недостатньо захищений канал «робот-смартфон» (напр., радіо-інтерфейс WiFi, взаємодія через Інтернет тощо). Програмний інтерфейс (API) смартфона розроблено у середовищі з відкритим кодом "Node.js" і оформлено у вигляді архіву APK ("Android Package Kit") з файлами програм під Android/iOS за допомогою популярного веб-сервісу "Ionic Framework" SDK (Software Development Kit) з відкритим кодом і клієнтським інтерфейсом для створення крос-платформних мобільних та стаціонарних додатків під операційні системи iOS/Android на мовах програмування HTML, CSS, JavaScript тощо. Побудований вихідний код був скомпільований та розгорнутий у веб-середовищі "Android Studio IDE" [104].

У 2023, D. Sufiyan та ін. (Singapore Univ. of Technology and Design) розробили гнучкий 3-модальний безпілотний літальний апарат (БПЛА), здатний працювати як робот з роторним крилом (rotor-wing), крейсерський робот (cruise robot), або робот з вертикальним зльотом/посадкою на хвіст (tailsitter) і нахилом вперед для горизонтального польоту, на відміну від вертикального зльоту/посадки (VTOL) з горизонтально орієнтованими фюзеляжами [105].

Робота [105] надає аналітичний огляд мультимодальних БПЛА, монокоптерів та різних режимів польоту, а також власну концепцію: 1) математична модель динаміки БПЛА на основі теорії «кватерніонів» (4-компонентних числових об'єктів, що розширюють поняття 2-компонентного комплексного числа; 2) апаратне/програмне проектування; 3) контролер орієнтації для переходу літака з режиму хвостового крила до крейсерського польоту; 4) динамічна структурно-параметрична оптимізація контролера орієнтації; 5) моделювання, результати та експериментальна оцінка мультимодальних процесів літака; 6) опис прототипу; 7) льотні випробування в контрольованому/відкритому середовищі.

В цій роботі продемонстрована здатність прототипу плавно переходити між режимами в повітрі. Зазначено, що польоти на відкритому повітрі проводилися в напівавтоматичному режимі зі стабілізацією положення, і майбутня робота передбачає реалізацію контролю положення на відкритому повітрі незалежно від контроля руху. Зроблено висновок, що повністю автономні польоти/переходи можуть виконуватися; хоча плоский аеродинамічний профіль, який використовувався в роботі, може бути неоптимальним для польоту з фіксованим крилом, будучи предметом подальшого вдосконалення польоту в крейсерському режимі [105].

У 2023, Іван Цмоць (НУ «Львівська політехніка») та ін. запропоновали інтегровану концепцію створення мобільних роботизованих платформ (MRP), що охоплює навігацію, попередню обробку даних та інтелектуальне управління на основі нечіткої логіки і сучасної елементної бази FPGA для обробки/розпізнавання даних в умовах перешкод та неповноти інформації [106].

Того ж 2023 року, фахівці з Португалії та Бразилії B. Stefanuto та ін. представили відкриту веб-платформу для дистанційного керування маніпулятором у рамках віртуальної академічної освітньої лабораторії з робототехніки, яка забезпечує 3D-візуалізацію моделі робота, інтерактивні кнопки-команди для віртуального маніпулювання реальним роботом, а також домашню сторінку автентифікації користувача для безпечного доступу [107].

У 2023, австралійські інженери Md. Moniruzzaman та ін. (Edith Cowan University) удосконалили генеративну нейронну мережу “cGAN” (Conditional Generative Adversarial Net – Умовна Генеративно-Змагальна Мережа) для управління наземних роботизованих платформ на кшталт відомої генеративної мережі Pix2Pix/cGAN [108].

Класична GAN генерує дані випадковим чином із навчального набору, натомість, умовна GAN (cGAN) додає умову (condition) *генератору* щоб навчити його генерувати дані відповідно до цієї умові, а також *дискримінатору* для оцінки того, чи згенеровані дані не лише реалістичні, але й відповідають заданій умові. окрім того, cGAN містить навчання через аналіз входів-виходів в процесі розпізнавання образів (Input/Output Image Mapping).

Популярна реалізація мережі cGAN відома як “Pix2Pix/CGAN”; її сутність у тому, що вона «навчилася» і вміє «перекладати зображення» з однієї області в іншу, але з іншими властивостями або стилем. У навчанні Pix2Pix використовують «міру індексу структурної подібності» SSIM (Structural Similarity Index Measure) для оцінки візуальної якості зображень, щоб зберегти структурні деталі об’єктів та покращити загальну якість зображення прогнозованих кадрів. Вона визначає ступінь подібності між двома зображеннями, де одне є оригіналом (еталоном), а інше – його версією, що була спотворена, стиснута чи відновлена. Загадане вище удосконалення мережі cGAN стосується додаткового фільтруючого алгоритму для видалення шуму з щільних компонентів оптичного потоку, що виникає через невідповідність частоти кадрів. [108]. Того ж 2023 року, в університеті м. Lleida (Іспанія) створено пристрій ARP-2 в межах проекту персонального робота-асистента для керування гуманоїда жестами людини з можливостями невербальної та вказівної комунікації [109]. Китайські вчені T. Jiang та ін. (Beijing Inst. of Automation) створили безпілотний наземний UGV (Unmanned Ground Vehicle) з підвищеним навантаженням, ROS-API та опційними інтерфейсами TTL, RS232/485, USB, Ethernet, CAN (Controller Area Network) для зв’язку хост-системи (МК, ПК, тощо) з електричною шиною CAN [110].

У січні 2024, M. Rybczak та ін. (Gdynia Maritime Univ., Poland) представили огляд зі 104 посиланнями на методи машинного навчання (ML) з учителем, без учителя та з підкріпленням (SL/UL/RL) щодо керування мобільними роботами, в якому розглянуто 8 класів робото-технічних ML-додатків: оцінка позиції, картографування середовища, одночасна локалізація та картографування (SLAM), класифікація місцевості, уникнення перешкод, слідування за шляхом, навчання ходьбою та координація кількох роботів. Окреслено відкриті проблемні питання: складність алгоритмів ML порівняно з обмеженими бортовими обчислювальними ресурсами; керування рухом робота та прийняття рішень у режимі реального часу; адаптивність робота до змінних середовищ; збір та обробка великих даних; надійність та безпека; розробка ML для роботів що природньо рухаються [111].

У лютому 2024, S. Zhang і H. Zhao (Shanghai Inst. of Technology, China) продемонстрували макет керування роєм мобільних роботів через робото-технічну операційну систему (ROS) на базі конструктора TurtleBot з відкритим API, розробленого на веб-платформі Gazebo з інструментом візуалізації Rviz, а також оптимізатором локального шляху TEB (Timed Elastic Band) для автономної багато-точкової навігації у відомому середовищі. Побудовано розподілену комунікаційну платформу для керування роєм роботів, одночасної локалізації та картографування (SLAM). Формування та відстеження рою реалізують TF-tools (програмні утиліти екосистеми ROS для відстеження кількох координатних систем змінних у часі). Найбільш відома реалізація TF-tools – бібліотека TF та її наступник TF2 [112].

У травні 2024, в Китаї відбувся 3-й Міжнародний симпозіум з управління та робототехніки з 93-ма академічними доповідями у трьох секціях (інтелектуальні системи, автоматизоване керування та інтелектуальне виявлення, роботи/планування шляхів). Серед них відзначимо алгоритм ORB-SLAM, метод покращення нечіткого зображення під водою, навігаційну систему для безпілотного підводного апарату та деякі інші [113].

.

У червні 2024, американські фахівці A. Raz та ін. опублікували структуру навчання з підкріпленням на основі т. зв. «зрозумілого штучного інтелекту» XAI (Explainable Artificial Intelligence) що використовує адитивні пояснення Shapley (Shapley Additive Explanations – SHAP) у непрозорому прийнятті рішень, протестовану на прикладі аварійного спуску високошвидкісного аерокосмічного апарату [114].

У липні 2024, I. Geles та ін. (Університет Цюріха, Швейцарія) продемонстрували автономне керування польотом квадрокоптера безпосередньо з пікселів зображення, без явної оцінки стану та доступу до інерційного вимірювального блоку IMU (Inertial Measurement Unit), використовуючи комбінацію простору спостереження та дії для візуально-моторного роботизованого інтелекту. Завдяки цьому, складні обчислення переміщуються на потужний наземний ПК (або хмару), що розширює можливості реалізації складних алгоритмів та великих нейронних мереж у режимі реального часу [115].

Узагальнена функціональна схема дистанційного управління мобільною платформою Інтернету речей

Розглянемо принципи дистанційного управління одним чи кількома роботизованими платформами людиною-оператором базової станції в реальному часі. Мобільні системи безпілотників зазвичай використовують радіочастотні діапазони SubGHz та WiFi-зв'язок 2.4 GHz на операційній глибині, а також канали на базі 4G/5G LTE для дистанційного керування на великі відстані [116].

Канальний рівень WiFi 802.11 з типами фреймів [117], [118], [119], [120]:

- фрейми «керування», що використовуються в базовому наборі послуг BSS (Basic Service Set) ad-hoc мережі, що керується точкою доступу (AP), тобто для приєднання, виходу або зміни мережі WiFi;

- фрейми «керування» для очищення зони, захоплення каналу, підтримки несучої, підтвердження отримання;

- фрейми «даних» для передачі корисного навантаження.

На рисунку 2.1. показано структуру фрейму 802.11 {заголовок, тіло, трейлер}. Заголовок і тіло є змістовною частиною для зберігання в буфері при надсиланні/отриманні. Трейлер контроля кадру FC додається при надсиланні для перевірки цілісність на приймальній стороні. Деякі поля заголовка є необов'язковими. Двобайтове поле «Керування кадром» містить тип кадру, швидкість передачі даних, шифр тощо; це може впливати на інші поля (напр., поля адреси залежать від бітів ToDS/FromDS «до/від місця призначення»).

Рисунок 2.1 – Загальна структура фрейму 802.11

Поле «Тривалість/ID» містить значення NAV (Network Allocation Vector – вектор розподілу мережі) для обмеження часу доступу до середовища: режим «доступу без конкуренції» (NAV=32768 для всіх кадрів); «широкомовний /багатоадресний» (NAV=0, тобто конкуренний доступ до середовища може розпочатися після завершення кадру); «відправлення останнього фрагмента» (NAV = міжкадровий проміжок IFG + час підтвердження фрагмента); «нефінальний фрагмент» (NAV = віртуальний «запит на відправлення» RTS для зменшення колізій кадрів, тобто 2´час підтвердження + 3´IFG + час відправлення наступного фрагмента). Адреса_1 (одержувач) означає станцію, яка декодує радіохвилі в кадр 802.11 (наприклад, хост призначення обробляє тіло кадру як IP-пакет). Адреса_3 – це ідентифікатор BSS (MAC-адреса точки доступу WiFi). Адреса_4 необов'язково використовується бездротовими мостами (також відомими як бездротова система розподілу WDS). Загальний розмір тіла кадру варіюється від 0 до 2304/2304+8, якщо WEP увімкнено; максимальний розмір кадру становить 2342 байти: 30 (заголовок) + 2312 (тіло) без FC. 802.11ac розширює максимальний розмір кадру («блок даних протоколу MAC» MPDU) до 11454 байт для агрегації PDU верхнього рівня. Точка доступу WiFi контролює заголовки кадрів і оновлює вектор NAV [120], [121].

Використання різних кадрів 802.11 є досить складним; зокрема, передача інформації про корисне навантаження вимагає перехресної взаємодії між кадрами керування та даних. Традиційні мережі WiFi в основному використовуються для публічного або приватного бездротового доступу до Інтернету за відсутності активної електронної протидії. Натомість, керування бойовими дронами під час військових конфліктів потребує додаткових рішень. За цих міркувань, далі синтезовано спеціальну модель взаємодії системи для дистанційного керування мобільною платформою дрона.

Розглянемо узагальнену функціональну схему дистанційного управління мобільною платформою Інтернету речей, яку назвем “Mobile_Platform-Remote_Control” (MPRC) на рисунку 2.2. На ньому взаємодіють мобільна платформа (MP) і станція дистанційного управління (RC), якою керує пілот-оператор (людина) або віртуальний AI-пілот. Мобільна платформа MP містить бортову сенсорну систему (для спостереження за поточним станом мобільної платформи MP та зовнішєю обстановкю), а також систему управління (для реалізації місії MP).

В режимі тренажера мобільна платформа MP симулюється у середовищі віртуальної реальності з інтерфейсом для реального або штучного пілота.

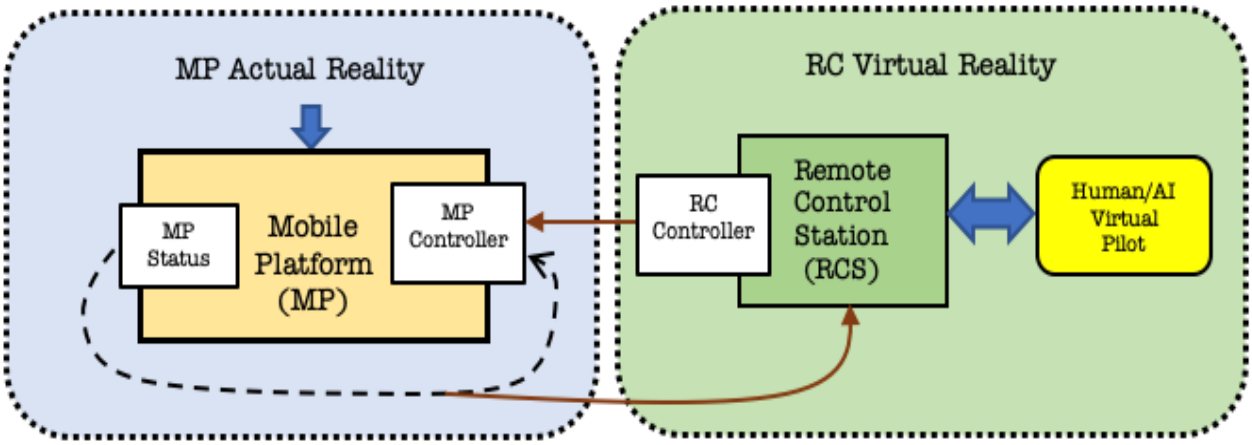


Рисунок 2.2 – Узагальнена функціональна схема дистанційного управління мобільною платформою Інтернету речей

Система управління має два входи: а) вектор внутрішнього стану мобільної платформи MP; б) зовнішній вектор від станції дистанційного управління RCS, яка контролюється реальним оператором або віртуальним пілотом зі штучним інтелектом. Мобільна платформа MP і станція RCS взаємодіють у режимі реального часу. Взаємодія MP-RCS здійснюється циклічно; кожен цикл T містить 4 фази: спостереження (Δt_1), повідомлення (Δt_2), аналіз (Δt_3) та команда (Δt_4); T=sum (Dt_1 , Dt_2 , Dt_3 , Dt_4).

Спостереження означає отримання набору даних про поточне середовище та внутрішній стан мультиплексного літака, враховуючи реакцію мультиплексного літака на нещодавні команди від станції дистанційного керування RCS.

Отриманий набір даних про поточний стан мобільної платформи MP та її зовнішнє оточення потім надсилається пакетними повідомленнями на станцію дистанційного керування RCS для оновлення загальної картини віртуальної реальності та місії в цілому, – як для людини-оператора, так і для віртуального пілота зі штучним інтелектом.

Пілот (людина та/або штучний інтелект) аналізує поточний перебіг подій у місії MP і активує відповідні команди зворотного зв'язку через інтерфейс пілота станції дистанційного управління.

Архітектура роботизованого мобільного комплексу Інтернету речей

Відповідно до рекомендацій I-IRA (“Industrial Internet Reference Architecture”), представимо інтерфейси між мобільною платформою MP і станцією дистанційного управління RCS (див. рис. 2.2) у вигляді 3-рівневої структури роботизованого комплексу Інтернету речей (Robotics Interoperability Framework, RIF), як показано на рисунку 2.3.

Схема RIF описує взаємодію функціональних компонентів: BCS (Basic Control System), MP (Mobile Platform) та RF-модемів на трьох рівнях: прикладному, канальному (Data Link Layer - DLL) та фізичному [38].

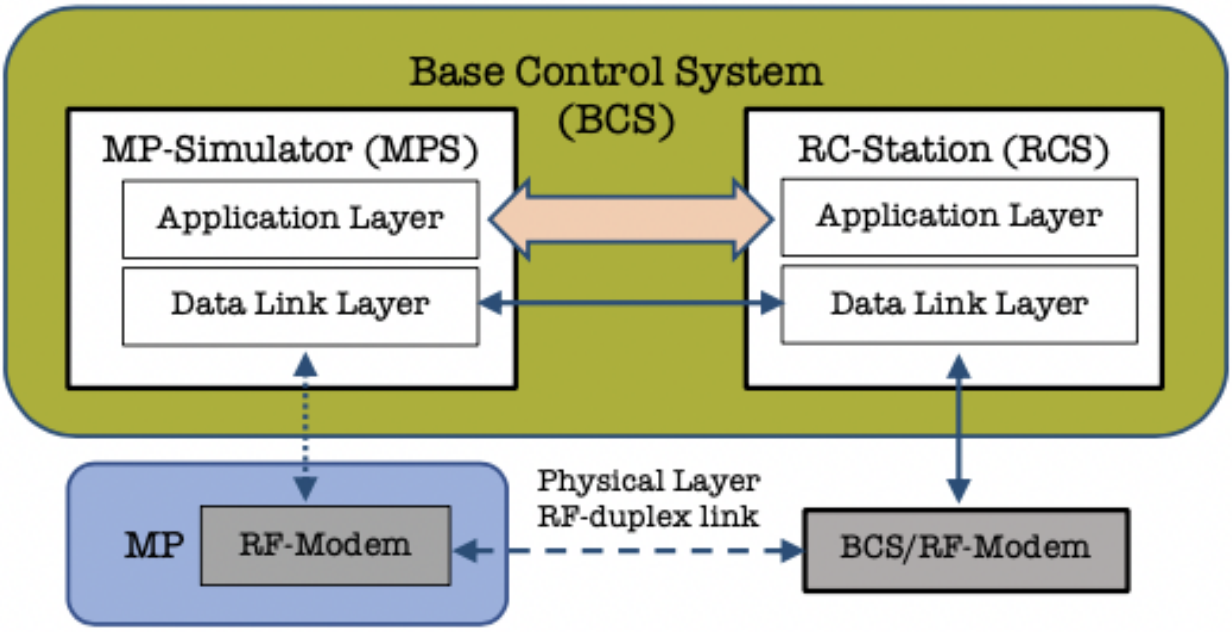


Рисунок 2.3 – Структура інтерфейсів роботизованого мобільного комплексу Інтернету речей – Robotics Interoperability Framework (RIF)

Базова система керування (BCS) на рис. 2.3 включає дві підсистеми: симулятор мобільної платформи (MPS) та станцію дистанційного керування (RCS), які потенційно можуть взаємодіяти на одному з трьох рівнів (прикладному, канальному або фізичному). Перші два (прикладний і канальний) відбуваються всередині самої BCS в автономному режимі (без використання реальної мобільної платформи). При цьому BCS генерує набір даних "Стан" для оточення мобільної платформи та її внутрішнього стану.

Згідно з RIF, взаємодія MP-RCS на прикладному рівні здійснюється через доступ до спільних наборів даних "Стан" та "Команда" відповідно до чотирьох фаз кожного циклу керування, розглянутих вище (спостереження, повідомлення, аналіз та команда). Канальний рівень взаємодії MP-RCS здійснюється через кадри змінної структури (у т.ч., звичайні кадри WiFi {заголовок, тіло, трейлер}, що кодуються формальною граматикою над машини Тюрінга). Фізичний рівень взаємодії MP-RCS може бути організований як в автономному, так і в реальному режимі через різні RF-дуплексні канали (наприклад, інтерфейс WiFi з нестандартними кадрами “raw-socket”).

Структура роботизованого мобільного комплексу RIF (рис. 2.3) розділяє проблему створення робототехнічних комплексів BCS (Base Control System) на три взаємо-пов'язані завдання фізичного, канального і прикладного рівнів:

- розробка сигнально-кодових конструкцій фізичного рівня, стійку до природних і штучних завад (MP/RF-Modem, BCS/RF-Modem; RF-duplex link); розв'язанню цього завдання присвячено багато досліджень;

- розробка алгоритмів захищеного кодування канального рівня (окремі аспекти цього завдання розглядається в даній дисертаційної роботі);

- розробка станції дистанційного керування (RCS) на прикладному рівні, включаючи динамічну модель робототехнічного транспортного засобу, моделювання місії та інтерфейс віртуального пілота; ця сфера активно розвивається останніми роками.

Побудова формальної граматики кодування фреймів Ethernet/WiFi

Технологія проводових локальних мереж Ethernet та її бездротовий аналог WiFi широко використовуються на рівні доступу в сучасному Інтернеті речей (IoT). Стандартами Ethernet і WiFi опікуються переважно групи 802.3 та 802.11 міжнародного Інституту інженерів з електротехніки та електроніки IEEE. Історично виникли два незалежні базові стандарти Ethernet – DIXv2.0 (спільна розробка компаній DEC, Intel і Xerox, 1982 – Ethernet-2) та IEEE-802.3 (10BASE5, 1983 – Ethernet-3). Специфікація IEEE-802.3x (1997) об'єднує обидва базові стандарти Ethernet за принципом зворотної сумісності. Вона визначає структуру і параметри фрейму Ethernet, інваріантні до способу його передачі (*структурно-параметричний код*), а також його відображення у послідовність цифрових символів для передачі по каналу зв'язку – *потоковий код фрейму*.

Потоковий код фрейму Ethernet для передачі по каналу зв'язку містить складові структурно-параметричного коду, які вбудовані за правилами певної *формальної граматики* у послідовність цифрових символів заданого алфавіту для *лінійного кодування*, з урахуванням особливостей взаємодії передавача і приймача фрейму, а також властивостей лінії зв'язку.

Зокрема, потоковий код містить такі додаткові структурні елементи як захисний інтервал між послідовними фреймами для запобігання колізій в лінії зв'язку і делімітації кінця фрейму, преамбулу і стартовий байт для синхронізації приймача з вхідним потоком даних та фіксації початку фрейму, контрольну суму структурного коду фрейму для перевірки його коректного прийому.

Розглянемо основні принципи потокового і лінійного кодування фреймів в технології Ethernet на скрученій мідній парі з точки зору їх можливої реалізації на логічних засадах машини Тьюринга. Введемо поняття структурно-параметричного коду фрейму Ethernet (Frame Parameter Code, або FPC) як упорядковану систему трьох векторів:

$$FPC = \{ [AD(DMAC, SMAC)],\ [ET()],\ [D(DL, DAP)] \},$$

де AD – фізичні адреси DMAC, SMAC отримувача та відправника фрейму (2×6 байт); ET – тип фрейму (2, 5, 6, 10 байт) в залежності від значення перших двох байт, а також п'ятого байту; D – корисні дані розміром DL байт в області пам'яті процесора з адресом доступу DAP (Data Access Point).

Код FPC, разом з контрольною сумою (4 байти) у потоковому коді фрейму 802.3x, повинен мати розмір 64÷1518 байт, (RFC 894, 1984). Звідси, $|SPF|+4 \le 1518$, або $(2\times6+|ET|+DL)+4 \le 1518$. Поле даних обмежено зверху: $DL \le 1502-|ET|$. При $|ET|=2$, маємо $0 \le DL \le 1500$; при $|ET|=10$ – $(0 \le DL \le 1492)$ байт.

Розглянемо три формати FPC в залежності від розміру поля ET.

1) $|ET|=2$ байти; це відповідає фрейму DIXv2.0 (Ethernet-2), а значення $ET=1536\div65534$ кодують тип фрейму, [3, с. 1]).

2) $|ET|=5$ або 6 байт; це відповідає фрейму IEEE-802.3 (Ethernet-3), де поле ET розширено у поле даних D на 3 або 4 байти заголовку LLC (специфікація IEEE 802.2). При цьому перші два байти у полі ET визначають кількість байт корисних даних $DL=0\div1497$ або 1496 байт. В потоковому коді, поле даних обмежено знизу: $DL \ge (64-18)=46$ байт; інакше воно доповнюється довільно до 46 байт.

Два молодші біти «CC» у п’ятому байті розширеного поля ET фрейму IEEE-802.3 (Ethernet-3) визначають один з трьох режимів протоколу LLC:

U (Unnumbered frame, CC=11) – датаграмний режим передачі фреймів як незалежних пакетів канального рівня, LLC=3 байти, |ET|=2+3=5 байт;

I (Information frame, CC=00, 10) – передача фреймів мультиплексними потоками по логічним з’єднанням з номерами LLN (Logical Link Numbers) у шостому байті розширеного поля ET; LLC=4 байти, |ET|=2+4=6 байт;

S (Supervisor frame, C=01) – фрейми службових повідомлень, що містяться у шостому байті розширеного поля ET; LLC=4 байти, |ET|=6 байт.

3) |ET|=10 байт; цей випадок відповідає специфікації SNAP Ethernet IEEE-802.3x (Sub-Network Access Protocol), фрейм якої містить у перших двох байтах розширеного заголовку ET кількість байт корисних даних DL=0÷(1500–8)=1492. При цьому, розмір DL у потоковому коді фрейму має бути не менше ніж (46–8)=38 байт, інакше воно доповнюється до 38 байт.

На відміну від фрейму IEEE-802.3 (Ethernet-3), перші 3 байти LLC-заголовку в розширеному полі ET фрейму SNAP Ethernet містять так звану LLC-заглушку (dummy LLC) з фіксованим значенням $(AA`AA`03)_{16}$ або $(AB`AB`03)_{16}$. Перші 3 з наступних 5 байт розширеного поля LLC (заголовку SNAP) містять 24-бітовий OUI (IEEE Organizationally Unique Identifier), який ідентифікує виробника і продавця мережевого обладнання. Останні 2 байти містять 16-бітовий номер типу фрейму (аналогічно полю ET в заголовку фрейму DIXv2.0). Розширення заголовку LLC було історично обумовлено поступовим збільшенням типів фреймів, попри обмежений 6-бітовий код типу у складі 3-го байту попередньої версії LLC. Оскільки значення $(AA`AA)_{16}$ та $(AB`AB)_{16}$ для перших двох байт LLC були раніше незадіяними (резервними), тому їх використали як «команди» розширення LLC заголовку. Намагання забезпечити сумісність існуючого мережевого обладнання з новими розробками призвели до надмірного ускладнення протоколу LLC. Через це, перша опція 1 (DIXv2.0 Ethernet-2) об’єднавчого стандарту IEEE-802.3x наразі виявилася більш популярною, ніж опції 2 і 3 (802.3 Ethernet-3 та SNAP Ethernet).

Характерними типами даних у фреймі Ethernet-2 DIXv2.0 є наступні: ET= $(0800)_{16}$ – у полі даних пакет протоколу IPv4; ET =$(0806)_{16}$ – пакет протоколу ARP (Address Resolution Protocol); ET = $(86DD)_{16}$ – пакет IPv6; ET = $(8847)_{16}$ – додатковий заголовок MPLS в полі даних; ET = $(88F7)_{16}$ – у полі даних пакет PTP (Precision Time Protocol); ET = $(B7EA)_{16}$ – у полі даних управляючі команди протоколу GRE (Generic Routing Encapsulation). Обладнання за уніфікованим стандартом IEEE 802.3x (1997) уможливлює сумісне використання фреймів обох базових стандартів Ethernet (DIXv2.0 та IEEE 802.3).

Структурно-параметричний код FPC містить повний набір параметрів взаємодії кінцевих абонентів. Коди FPC формуються на канальному рівні OSI взаємодіючих сторін, і тимчасово зберігаються в оперативній пам'яті адаптерів Ethernet з метою їх подальшої передачі на фізичний рівень лінії зв'язку, або навпаки, – драйверам мережевого рівня власного вузла.

Потокове кодування фреймів Ethernet/WiFi

Цифровий код FPC не може безпосередньо передаватися в лінію зв'язку за різних технологічних обставин; натомість він має бути перетворений у послідовність цифрових символів лінійного коду – *потоковий код фрейму* (Frame Streaming Code, або FSC) з додатковими структурними елементами: міжфреймовий захисний інтервал IFG (Inter Frame Gap), преамбула синхронізації (PRB), стартовий байт SFD (Start Frame Delimiter), та контрольна сума CRC (4 байти) від векторів (AD, ET, D) структурно-параметричного коду фрейму FPC.

Особливістю потокового кодування є обмеження мінімального розміру FSC заради того, аби відрізнити його від сміття в лінії при колізіях. Тому вектор корисних даних D коду FPC перетворюють у змінне поле даних PL (Payload) потокового коду FSC, а його змістовну частину (AD, ET, PL, CRC) тримають не менше 64 байт (RFC 894).

При цьому мінімальний загальний розмір потокового коду фрейму Ethernet (без захисного інтервалу IFG) становить 72 байти:

$$|FSC|_{min} = |PRB|_7 + |SFD|_1 + |DMAC|_6 + |SMAC|_6 + |ET|_2 + |PL|_{46} + |CRC|_4 = 72.$$

Мінімальний розмір поля даних, разом з заголовком LLC, становить $|PL|_{min} = 64–(6+6+2+4) = 46$ байт. Якщо вектор корисних даних D і заголовок LLC разом менше 46 байт, то поле PL доповнюється баластом до 46 байт (padding), наприклад, нулями та одиницями.

Для фрейму DIXv2.0 (Ethernet-2) без LLC, мінімальний «чистий» вектор D складає 46 байт. Фрейм 802.3 з 4 байтами LLC, має «чистий» вектор D від (46-4)=42 байт. У фреймі SNAP Ethernet 802.3x (3 байти LLC + 5 байт SNAP), мінімальний «чистий» вектор D складає (46–3–5)=38 байт.

На приймальній стороні каналу, в потоковому коді FSC виділяється FPC між стартовим байтом SFD і захисним між-фреймовим інтервалом IFG, тому поле даних PL детерміновано локалізується між заголовком у 14 байт (два MAC адреси по 6 байт плюс поле типу даних 2 байти) та кінцевим полем контрольної суми CRC (4 байти).

Натомість, потоковий код фрейму DIXv2.0 (Ethernet-2), в якому поле даних PL має мінімальним розмір 46 байт, не вказує безпосередньо на обсяг корисних даних D (оскільки може мати місце padding). У цьому випадку, відокремлення «чистих» даних від баласту у полі PL має вирішуватися на наступному (мережевому) рівні OSI, за допомогою параметрів заголовку вкладеного пакету (наприклад, IP-пакету).

Навпаки, у фреймі IEEE-802.3 (Ethernet-3), поле типу має значення DL – фактичного розміру корисних даних D, розміщених у полі даних PL одразу після LLC заголовку. Як вказано в розділі 1, структура і розмір LLC заголовку та його можливого розширення у фреймі SNAP Ethernet, визначаються першими трьома байтами LLC. Це дозволяє однозначно ідентифікувати корисні дані D у полі даних PL незалежно від розміру цього поля.

Слід зазначити, що застосування двох різних типів корисного навантаження фрейму Ethernet-2 (при інкапсуляції IP-пакетів за протоколами версій IPv4 $(0806)_{16}$ та IPv6 $(86DD)_{16}$) є надлишковим, оскільки перші чотири біти у заголовку будь якого IP-пакету визначають конкретну версію IP-протоколу, а саме: «0100» – версія IPv4; «0110» – версія IPv6.

Обмеження потокової структури фрейму FSC мінімальним розміром у 72 байти (без IFG) з’явилося у ранніх стандартах локальних мереж, в яких декілька адаптерів Ethernet підключались до спільної шини (коаксіального кабелю). В таких мережах, при короткому фреймі активного адаптера-передавача та певній довжині лінійного кабелю, інші адаптери можуть спостерігати хибну паузу в лінії, і почати передачу в лінію власних фреймів, не дочекавшись кінця передачі активного адаптера. Виникає колізія одночасної генерації в лінію декількох фреймів від різних джерел, яка спотворює лінійний сигнал.

Включення елементів фрейму (IFG≥12 + PRB=7 байт + SFD=1)·байт у потоковий код FSC обумовлено особливостями ранніх версій Ethernet що досі підтримуються багатьма сучасними стандартами Ethernet різних поколінь. Окремі адаптери Ethernet підтримують різні набори можливої швидкості прийому/передачі цифрових сигналів (10, 100, 1000 Мбіт/с і вище).

При цьому, передача кожного фрейму Ethernet переважно здійснюється окремою незалежною сесією у непередбачувані проміжки часу; всередині кожної сесії, прийом/передача фрейму є синхронним процесом, тобто тактовий генератор приймача має чітко відслідковувати періодичні часові інтервали для детектування окремих лінійних символів вхідного потоку даних.

Захисний інтервал IFG розділяє послідовні фрейми в лінії зв’язку, преамбула забезпечує частотну і фазову синхронізацію приймача з вхідним цифровим потоком, а байт SFD сигналізує про початок заголовку фрейму.

Натомість, сучасні локальні мережі Ethernet будуються переважно на комутаторах і дуплексних лініях зв’язку, де кожен передавач генерує лінійний сигнал одному приймачу, що унеможливлює інтерференцію лінійних сигналів від декількох передавачів.

Передові технології, в принципі, дозволяють перманентно зберігати фізичну і логічну синхронізацію взаємодіючих сторін в каналі зв’язку, що є підґрунтям для перегляду принципів потокового кодування фреймів в каналах Ethernet в напрямку зменшення його надлишковості.

Потокове кодування даних на фізичному рівні

Потоковий код FSC (Frame Streaming Code) для передачі в лінії формується на основі структурно-параметричного коду (FPC), відповідно до протоколів фізичного рівня і типів кабелю. На фізичному рівні Ethernet, популярними є бінарні лінійні коди – з двозначними фізичними сигналами (Mii-коди) або тризначними (MLT-3, 100 BASE-TX), [4]. В сімействі Base Ethernet (10Base5, 10Base2, 10BaseT, 100BaseTX, 1000BaseT – окрім 10Base36), не застосовують гармонійну несучу для сигналізації символів, натомість лінійний сигнал генерується безпосередньо у смугу пропускання каналу (“baseband signal”). Простим бінарним MII-кодом є код NRZ-PAM2 з високою завадостійкістю і малою смугою частот, популярний у SerDes комунікаціях (**Ser**ialization- **Des**erialization – перетворення паралельного коду у послідовний і навпаки), а також в оптичних мережах доступу. Його недолік – постійна складова сигналу і вади синхронізації при монотонних послідовностях однакових символів.

Інтерфейси Ethernet-10Мбіт/с запроваджують бінарний само-синхронізований інверсний манчестерський код (MC-code) “IEEE 802.3 convention”, в якому кожен біт є водночас тактовим сигналом (оригінальний MC-код має нотацію “G.E. Thomas convention”).

Інтерфейси Ethernet 100BASE-T4/T1/1000BASE-T1 використовують блочний код 8B6T у 3-символьному алфавіті PAM3. Так, специфікація 100Base-T4 (стандарт IEEE 802.3u) застосовує кабель «4 кручені пари 3-ї категорії»; дві пари спрямовані протилежно, а інші дві перемикаються в напрямку передачі. В кожен момент часу, три з чотирьох пар передають дані, а одна прослуховує лінію на предмет колізій. За кожні 2 такти, три кручені пари передають блок з 6 трійкових символів ($3^6 = 729$ значень), який кодує 256 значень октету.

Інтерфейс 1000BASE-T реалізує код PAM5, в якому 2 позитивних і 2 негативних рівні амплітуди за кожний такт кодують 2 біти інформації (4-символьний алфавіт), а нульовий сигнал є надлишковим (не використовується). Інтерфейс 2.5GBASE-T застосовує код PAM16 [122], [123].

Принципи побудови формальної граматики для кодування фреймів

В цілому, цифровий потік пакетної IP-мережі на різних рівнях фєрархії протоколів (згідно деякої еталонної моделі взаємодії відкритих систем), можна представити в термінах багаторівневої формальні граматики деякого порядку. Теорія формальних граматик узагальнює і формалізує інтуітивну логіку побудови звичайних граматик для мови людського спілкування. Прикладами формальних граматик є деякі математичні теорії, мови програмування (C, Python, тощо), інтерфейси взаємодії з автоматичними пристроями і т. д.

Формальна граматика (позначимо її **FG**) базується на «абстрактному алфавіті» **A** (набір символів, який, може мати не тільки «літери», але й спеціальні «синтаксичні» знаки типу «пробіл»). Формальна граматика **FG** має свій «синтаксис» (**SIN)** – набір правил побудови окремих «слів» і «речень» з «літер» синтаксичних знаків алфавіту **A**. Серед множини «слів» граматики, виділяють «зарезервовані слова» («команди»), які утворюють «тезаурус» (словник) граматики Всі інші (незарезервовані) «слова» відіграють роль параметрів і даних, застосування яких має відповідати правилам синтаксису.

Формальна граматика **FG** має реальний сенс («семантику» **SEM**), якщо існує «дехто» чи «дещо» – такі що «розуміють правильні тексти», тобто, вміють перетворювати їх у конкретні дії (напр., математик, інженер-технолог, програма-компілятор, програмований станок-автомат). Важливо, щоб «семантика» формальної граматики підтримувала діагностику помилок у відповідниї «текстах». Формальну граматику **G** можна представити як трійку понять (алфавіт, синтаксис, семантика): **G**:= {**A**, **SIN**, **SEM**}.

По аналогії з відомою шкільною граматикою, виділимо такі граматичні поняття у контексті ієрархії цифрових структур даних: *алфавіт*, *орфографія*, *морфологія*, *синтаксис*, *словосполучення, пунктуація, речення, повідомлення, розділ документа, файл документа*. «Літери» утворюють «слова» згідно орфографії. Слова утворюють форми морфології, які об'єднуються у словосполучення і речення згідно правил синтаксису.

Послідовність речень утворює повідомлення різних розмірів, типів і семантичних рівнів ієрархії за правилами пунктуації – розділи документа, файл документа тощо [124]. На канальному рівні OSI в IP-мережах роль вищезгаданих граматичних речень в межах деякої багаторівневої формальної граматки (позначимо її $\mathbf{FG^N}$) відіграють фрейми (наприклад, фрейми Ethernet, WiFi).

Мережевий рівень OSI обробляє IP-пакети, за допомогою яких здійснюється передача змістовної інформації – від коротких повідомлень до об'ємних файлів документів. При цьому, достатньо великий файл розділяється для передачі по IP-мережі на окремі IP-пакети. Наприклад, у найбільш поширеній сьогодні версії протоколу IPv4, розмір пакету (разом с заголовком) у байтах визначається значенням поля "Total Length" (16 біт), і його максимально можливе значення становить $(2^{16}–1)=65535$ байт.

Згідно стеку TCP/IP, IP-пакети вкладаються у фрейми канального рівня OSI, зокрема, у фрейми WiFi, в яких корисне навантаження, зазвичай, обмежено розміром 1500 байт. Тому, при передачі пакетів значних розмірів по IP-мережі вони розділяються на окремі фрагменти.

З точки зору формальної граматики $\mathbf{FG^N}$, будемо вважати ці фрагменти IP-пакетів «*повідомленнями*», а самі IP-пакети що фрагментуються – «*розділами документу*». Далі по ієрархії вгору, з окремих «розділів документу» будемо розглядати «файл документу». На верхньому (семантичному) рівні граматики $\mathbf{G^N}$ будемо розглядати виконавців (люди-суб'єкти, об'єкти, драйвери, пристрої), які здатні інтерпретувати (розуміти, відображати, виконувати, реалізовувати) зміст файлового документу. Зведемо вказані граматичні форми у таблицю 5.

Таблиця 2.1 – Граматичні форми пакетної передачі даних в IP-мережі

Рівень	Граматична форма	Цифрова форма	Протокол
7. Прикладний	Виконавець	Драйвер	API
6. Транспортний	Документи	Файли	TCP
5. Мережний	Розділи документу	IP-пакети	IP
4. Мережний	Повідомлення	IP-фрагменти	IP
3. LLC	Речення	Фрейми	Протокол безпеки WiFi
2. MAC	Символи	Лінійні коди	Модем
1. Фізичний	Знаки	Ел. сигнали	Трансівер

Побудуємо трирівневу формальну граматику **FG³** (знаки, символи, речення - див. таб. 5) для потокового кодування фреймів FSC (Frame Streaming Code) в каналі Ethernet/WiFi. Прототипом фізичного рівня візьмем блочний код 8B6T з *трійковим лінійним кодуванням* PAM3, який здійснює взаємодію суміжних адаптерів Ethernet 100BASE-T4 в режимі «частковий полу-дуплекс» з динамічним розподілом ємності каналу в протилежних напрямках. Сучасні лінії реалізують трійковий код також іншими способами модуляції.

Сформулюємо п’ять правил формальної граматики **FG³** для потокового кодування фреймів FSC (Frame Streaming Code) в каналі Ethernet/WiFi.

Формальна граматика **FG³** (алфавіт + синтаксис)

1) Алфавіт граматики **FG³** («_», 0, 1) має *синтаксичний знак* «пробіл» і дві «літери» («0», «1»).

2) Послідовність пробілів синтаксично еквівалентна одному пробілу.

3) «Словом» у **FG³** є послідовність «літер» відокремлена «пробілами». «Ключове слово» у **FG³** є «слово» не більше 3 «літер»; інші «слова» є даними.

4) Коди потокових даних фрейму FSC передаються «пачками» FSC_n в окремих сесіях, що розмічаються тегами «пробіл» і парою скобок −{...}− без використання захисного між-фреймового інтервалу (IFG), преамбули (PRB) та стартового байту фрейму (SFD) у вигляді послідовності

$$-\{...\}-\{\ FSC_1, FSC_2,..., FSC_N\}-\{...\}-,\ N\geq 0.$$

5) Потоковий код FSC формується з трьох складових частин:

$$FSC=[AD(...);\ DAT(...);\ CRC(...)],$$

де AD – *адресний вектор* (DMAC, SMAC); DAT – *вектор даних* (Type, Payload); CRC – *контрольна сума* від векторів AD і DAT.

Правила 1÷5 визначають алфавіт і синтаксис трирівневої формальної граматики **FG³** для потокового кодування фреймів відповідно до структури інтерфейсів роботизованого мобільного комплексу Інтернету речей (рис. 12), і представляють собою удосконалену модель динамічного кодування даних мобільної ad-hoc мережі IoT.

2.2. Розробка прикладної машини Тьюринга для структуризації даних в каналі зв’язку

Логіко-математична концепція «Машини Тьюринга» (МТ) має важливе теоретичне і опосередковане прикладне значення у різних галузях знань, у т. ч., для захисту даних в комп’ютерних системах та мережах, оскільки вона визначає принципову можливість обчислення та межу ефективності відповідних алгоритмів, включно з криптографічними.

Теза Черча-Тьюринга (Church-Turing thesis) стверджує, що будь-яка обчислювальна задача, яка може бути вирішена алгоритмічно (тобто, поетапною процедурою), може бути вирішена і машиною Тьюринга. Існує універсальна машина Тьюринга (Universal Turing Machine), яка може імітувати роботу будь-якої іншої машини Тьюринга. Ця теза обґрунтовує модель програмованого комп'ютера, де одна машина може виконувати різні програми, закодовані на її стрічці пам’яті ([125], с. 78).

З цього витікає, що будь-який алгоритм кодування чи захисту даних, може бути виконаний універсальною МТ. Це забезпечує теоретично гарантовану можливість обчислити ці методи за обмежений час, що є базовим принципом криптографії. Також, МТ використовується для оцінки криптографічної надійності. Система вважається надійною, якщо найефективніший злам вимагає неприйнятно великої кількості кроків МТ (експоненційно багато часу), що робить злам нереальним. Нарешті, МТ дозволяє показати, що деякі завдання є алгоритмічно нерозв'язними в принципі (т. зв. Undecidable Problems).

Наприклад, неможливо строго довести, що програма шифрування є на 100% безпечною (т. зв. «Проблема зупинки» алгоритму). Тому не варто витрачати зусилля на пошук рішення такого типу завдань [126].

Математична модель машини Тьюринга ([127], травень 1936), разом із подібною до неї «машиною Поста» МР ([128], жовтень 1936) є одними з перших найбільш відомих теоретичних розробок в галузі скінченних автоматів (Finite State Automaton - FSA), яка почала активно розвиватися всередині минулого століття (Mealy G. H, 1955, [129]); Moore E., 1956, [130]).

На рисунку 2.4 показано функціональну модель абстрактного скінченного автомата FSA з виходом (FSA with output) типу «Машина Мілі» (“Mealy machine” - ММО), яка містить три основні структурні компоненти:

а) конечна «пам’ять» M (memory), що визначає внутрішній стан автомата $S(t)$ в поточний дискретний момент часу t ;

б) функціональний модуль «λ», який перетворює вхідні символи $x(t) \in A_X$ з алфавіту A_X у вихідні символи $y(t) \in A_Y$ з алфавіту A_Y в залежності від внутрішнього стану S(t): $x(t) \rightarrow y(t) = \lambda\{S(t), x(t)\}$;

в) функціональний модуль «δ» який обумовлює перехід автомата з його поточного стану S(t) в наступний стан S(t+1) в залежності від вхідного символу $x(t) \in A_X$ і внутрішнього стану S(t): $S(t) \rightarrow S(t+1) = \delta\{S(t), x(t)\}$.

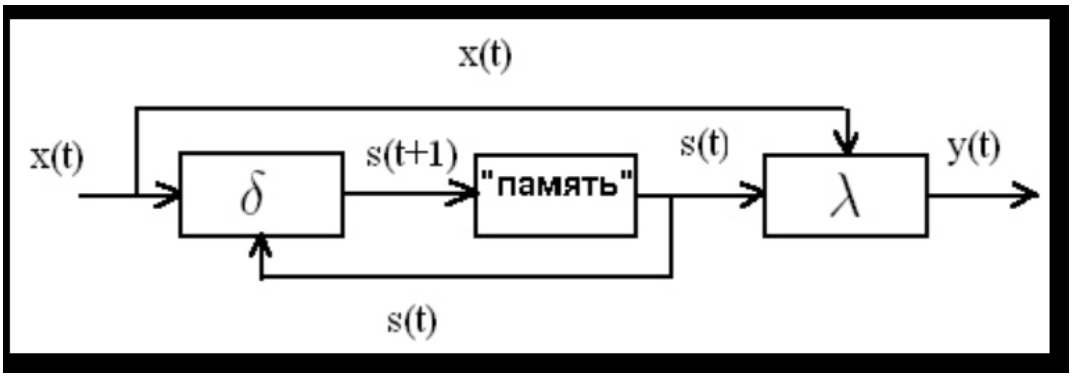


Рисунок 2.4 – Функціональна модель скінченного автомата типу «Машина Мілі з виходом» (ММО)

Множини символів з алфавітів A_X, A_Y можуть перетинатися між собою, або бути одним спільним алфавітом A. Функціональні модулі («δ», «λ») уособлюють деякий «алгоритм», що визначає конкретний тип автомата Мілі. Окремим випадком автомата Мілі є автомат Мура, де вихідний символ y(t) не залежить від вхідного символу x(t): $y(t) = \lambda\{S(t)\}$.

Абстрактна модель скінченного автомату FSA типу «Машина Мілі з виходом» (ММО, рис. 14) не конкретизує свої структурні компоненти та вхідний і вихідний алфавіти, а лише формулює загальні вимоги до їх властивостей:

- внутрішня пам'ять М автомата є конечною;
- комірка пам'яті М містить тільки один символ з алфавітів A_X або $A_{Y;}$
- функціональні модулі δ, λ мають конечні набори правил;
- алфавіти A_X, A_Y є конечними від одного і більше символів кожний;
- послідовність вхідних символів є необмеженою.

В цілому, скінченний автомат FSA типу ММО *не є суто формальним об'єктом*, оскільки він спирається на апріорно (неформально) визначені поняття функціональних модулів і абстрактних алфавітів.

Скінченний автомат ММО (рис. 14) можна порівняти з деяким контролером, який у кожний дискретний інтервал часу t перебуває у певному внутрішньому стані S_t, починаючи від нульового $S_0=S(t=0)$, і працює циклічно в режимі т. зв. з «відкритого циклу» – один цикл на кожний дискретний інтервал часу t. Дискретні інтервали часу визначаються послідовністю надходження вхідних символів, а також періодом виконання контролером усіх дій для циклу обробки одного вхідного символу. Якщо черговий вхідний символ відсутній, дискретний час t умовно «зупиняється» (контролер переходить в режим очікування наступного символу на вході).

У кожному циклі автомат ММО виконує три дії по обробці одного чергового вхідного символу:

- зчитується і аналізується черговий вхідний символ x(t);
- генерується черговий вихідний символ y(t) в залежності від x(t), S_t;
- змінюється внутрішній стан автомату: $S_t \rightarrow S_{t+1}$.

Якщо вхідний алфавіт A_X має лише один символ, автомат просто реагує на послідовність таких символів; при цьому, на кожному циклі внутрішній стан S_t , а також вихідний символ y(t), в загальному випадку змінюється (оскільки він залежить не тільки від входу, але й від попереднього значення S_{t-1}.

Машина Тьюринга (МТ) є підтипом скінченного автомата FSA «Машина Мілі з виходом» (ММО). Особливості МТ полягають у наступному.

1) Машина Тьюринга застосовує один спільний алфавіт **A** для вхідних та вихідних символів x(t), y(t), на відміну від машини Мілі ММО, де вхідний та вихідний алфавіти A_X, A_Y можуть бути різними або частково співпадати.

2) Машина Тьюринга у своєму (спільному для входу і виходу) алфавіті **A** може мати *від* двох і більше символів, на відміну від автомата Мілі що містить *від одного* і більше символів у кожному з вхідних та вихідних алфавітів A_X, A_Y. При цьому, найбільш актуальним варіантом машини Тьюринга є алфавіт з *трьома символами* **A**:={ «_», «0», «1»}, з яких «0», «1» є умовні «літери» для утворення «слів», а символ «_» – це умовний синтаксичний знак «пробіл» для відокремлення слів у послідовному потоці символів.

3) На відміну від *необмеженої у часі* «*по-символьної*» обробки вхідних даних автоматом FSA типу машини Мілі ММО (рис.14), машина Тьюринга (МТ) реалізує *«пакетну» обробку* символів за *обмежений період часу* T від t_0 до t_M. Кожен символ обробляється у вкладеному циклі (такті) на дискретному інтервалі часу $t_m \in [t_0, t_M]$, де M – конечна кількість тактів за період часу T.

4) На кожному періоді T пакетної обробки символів, машина Тьюринга отримує увесь пакет вхідних даних (послідовний конечний набір символів) у першій частині зовнішньої пам'яті (умовної «стрічки» з комірками для окремих символів); друга (необмежена) частина стрічки призначена для тимчасового зберігання проміжних результатів, а також для вихідних даних (символів). Відповідно, кожна комірка «стрічки пам'яті» МТ повинна мати 3 дискретні стани, на відміну від бітових елементів пам'яті традиційних комп'ютерів з двома станами, з яких складаються октети по 8 біт («байти»).

5) На кожному такті (дискретному інтервалі часу $t_m \in T$) машина Тьюринга зчитує деякий вхідний символ $x(t_m)$ зовнішньої пам'яті (стрічки), генерує і записує на стрічку черговий вихідний символ $y(t_m)$, а також змінює свій внутрішній стан $S(t_m) \rightarrow S(t_{m+1})$.

На рисунку 2.5 представлено функціональну схему машини Тьюринга (ТМ) як підтип скінченного автомата FSA типу «Машина Мілі з виходом» (ММО). Машина Тьюринга, окрім внутрішньої конечної пам’яті автомата Мілі ММО (рис. 14), має необмежену конечну зовнішню пам’ять у вигляді «стрічки з комірками пам’яті», яка «може розширюватися (вправо) скільки потрібно».

Комірки пам’яті на стрічці зовнішньої пам’яті МТ взагалі можуть мати два і більше можливих станів для відображення символів алфавіту А, але, в даному випадку, йдеться про найбільш популярну машину Тьюринга з три-символьним алфавітом («пробіл» «_» та дві «літери» «0», «1»).

На стрічці пам’яті МТ є дві зони: ліва (конечна) *пам’ять вхідних даних* x_0 ... x_n, яка заповнена на початку роботи МТ (позначена жовтим кольором); права (необмежена) операційна пам’ять проміжних та вихідних даних (білий колір). Доступ до комірок пам’яті на стрічці (читання/запис даних) здійснюється «головкою», яка контролюється блоком управління МТ шляхом переміщення головки на одну комірку вліво-вправо на кожному такті роботи МТ. Блок управління МТ є, фактично, скінченним автоматом Мілі ММО, який отримує вхідні символи конкретно шляхом їх зчитування зі стрічки, а також перетворює кожен вхідний символ у вихідний символ та/або переміщує головку за алгоритмом роботи функціональних блоків (δ, λ).

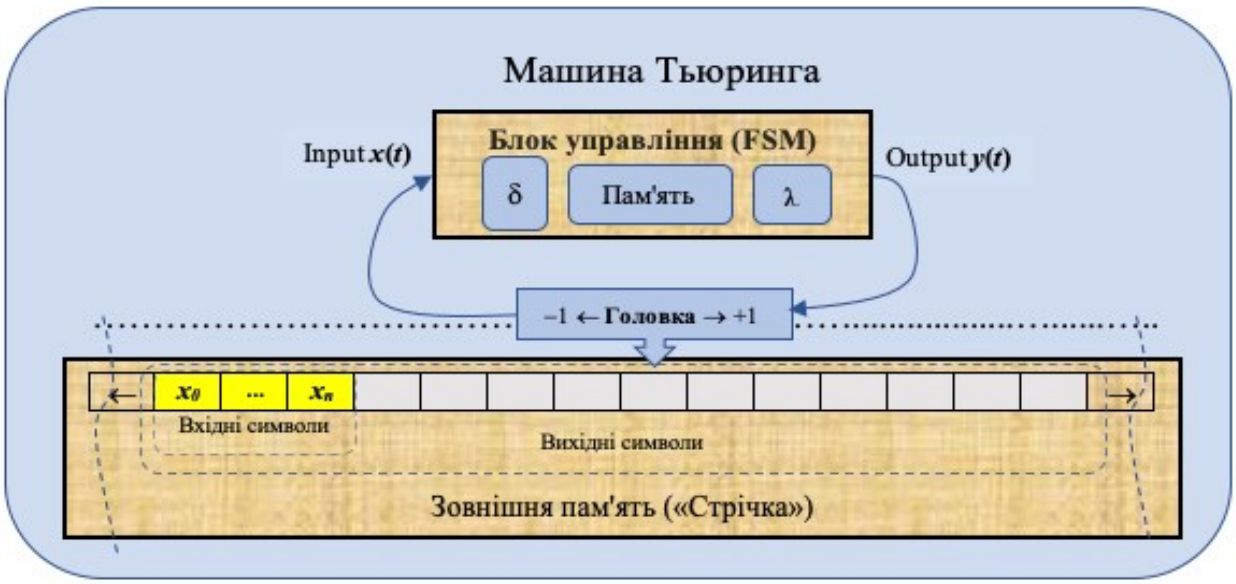


Рисунок 2.5 – Функціональна модель машини Тьюринга

Принцип роботи машини Тьюринга (рис. 2.5) полягає у наступному.

1) Початковий стан (Reset) в дискретний момент часу t_0 :

- Внутрішня пам'ять МТ знаходиться у стані S(0);

- В пам'яті вхідних даних стрічки записані символи вхідного алфавіту A_I, що використовуються функціями δ і λ алгоритму МТ як скінченного автомату Мілі (див. рис. 2.4);

- В операційній частині стрічки записано пусті символи вихідного алфавіту A_O);

- Головка встановлена на нульову комірку стрічки.

2) На поточному інтервалі часу *t* виконується наступне:

- Зчитується вхідний символ $x(t) \in A_{IN}$ з поточної комірки під головкою;

- Аналізується вхідний символ x(t) і поточний стан S(0), в результаті чого обчислюється вихідний символ y(t) і записується у поточну комірку стрічки: $x(t) \rightarrow y(t) = \lambda\{S(t), x(t)\}$;

- Головка переміщується на одну позицію вліво чи вправо;

- Змінюється (необов'язково) поточний внутрішній стан МТ:

$$S(t) \rightarrow S(t+1) = \delta\{S(t), x(t)\};$$

3) По завершенню процесу: результат знаходиться на стрічці пам'яті.

Абстрактний скінченний автомат Мілі працює необмежено у часі у за принципом «вхідний символ» $\rightarrow$ «вихідний символ» (подібно типовому контролеру IoT з відкритим «безкінечним» циклом обробки вхідних даних). «Скінченність» автомата Мілі полягає в конечності його внутрішніх станів і алгоритму (модулі δ, λ).

Натомість, машина Тьюринга (МТ) описує повний цикл пакетної обробки вхідного набору даних на стрічці зовнішньої пам'яті. Основним результатом циклу роботи МТ є вихідний пакет даних на стрічці.

Представлена вище функціональна модель машини Тьюринга є універсальним, але абстрактним логіко-математичним інструментом. Для практичного застосування, його необхідно конкретизувати у вигляді так званої «прикладної машини Тьюринга» (Applied Turing Machine, AMT).

Прикладна машина Тьюринга для гнучкого кодування фреймів.

Введемо поняття «Прикладна машина Тьюринга» (“Applied Turing Machine”, яку позначимо АМТ). Даний термін не є стандартним або офіційно визнаним у теоретичній інформатиці, але ідея, яку він описує, лежить в основі сучасної обчислювальної техніки, оскільки він логічно описує модель що має практичний, фіксований набір інструкцій (на відміну від абстрактної МТ).

Для створення прикладної машини Тьюринга АМТ на основі абстрактної функціональної моделі машини Тьюринга МТ з трьох-символьним алфавітом А = {«0», «1», «_»} необхідно додатково визначити:

- систему команд (відповідність мнемокодів команд їхнім «машинним кодам» у трійковому алфавіті);

- синтаксис (формальну граматику) для запису/використання команд;

- семантику (способів інтерпретації команд).

Визначимо систему команд прикладної машини Тьюринга (АМТ) як показано на рисунку 2.6. Застосуємо синтаксис формальної граматики FG, сформульований вище з п’яти правил потокового кодування фреймів FSC (Frame Streaming Code) в каналі Ethernet/WiFi (див. розділ 2.1).

№	Код	Мнемо	Семантика	Semantics
1	0	.	Точка	Dot
2	1	,	Кома	Comma
3	00	;	Крапка з комою	Semicolon
4	01	:	Двокрапка	Colon
5	10	(	Ліва дужка	Left Parenthesis
6	11	)	Права дужка	Right Parenthesis
7	000	H	Заголовок	Header
8	001	B	Тіло	Body
9	010	FC	Контрольна сума	Frame Check sequence
10	011	[	Ліва скобка	Left Bracket
11	100	]	Права скобка	Right Bracket
12	101	{	Ліва фігурна скобка	Left Brace
13	110	}	Права фігурна скобка	Right Brace
14	111	**$**	Розширення команд ≥ 4 біт)	Extension

Рисунок 2.6 – Система команд прикладної машини Тьюринга (АМТ)

Сформулюємо узагальнену формальну граматику FG прикладної машини Тьюринга (AMT) з правилами семантики потокового кодування фреймів (FSC).

Формальна граматика прикладної машини Тьюринга (AMT)

1. TM-слово – це непорожня послідовність літер «0» або «1», розділених одним або кількома «пробілами».

2. TM-команда – це TM-слово з 1÷3 «літер»; інші слова є TM-даними.

3. Два або більше «пробілів» еквівалентні одному; повторення тегів неприпустимо; дужки типу ( ), [ ], { } мають бути парними.

4. Вкладення тегів {{...}} або [[...]] заборонено.

5. Теги {...} визначають сеанс передачі об'єктів типу FPC, FDC тощо.

6. TM-фрейм TMF:=[(n), H(...); B(...); FC(...)], n – номер фрейму, H – заголовок; B – тіло фрейму з вектором даних програми (Тип, Корисне навантаження); FC – послідовність перевірки фрейму, розрахована для заголовка та тіла. Усі компоненти TMF можуть бути фіктивними, як і сам TMF.

7. Сеанс передачі фреймів TMF: {...}:= {TMF1, TMF2 ... }.

8. Послідовність сеансів TMF: “{...}{...}...{...}”, де теги «“», «”» є розширення системи команд через тег «$» у строчці 14 (див. рис.1, с. 74).

Наведений перелік з восьми синтаксичних правил використання тегів розмітки потоку, разом із правилами формування складових цифрового потоку у розділі 2.1, визначає прикладну машину Тьюринга (AMT) для потокового кодування фреймів в каналі зв’язку мобільної ad-hoc мережі Інтернету речей.

Значущість концепції машини Тьюринга для розробки алгоритмів потокового кодування даних полягає в тому, що її набір інструкцій не є статичним, а визначається набором правил (системою команд), яка може бути різною не тільки для кожної конкретної прикладної машини Тьюринга, але й динамічно змінюватися в процесі передачі окремих фреймів і повідомлень.

Далі, прикладна машина Тьюринга AMT використовується для побудови недетермінованої прикладної машини Тьюринга NdTM (“Nondeterministic Turing Machine”) для крипто-стійкого динамічного кодування даних ad-hoc мережі Інтернету речей.

2.3. Синтез методу крипто-стійкого динамічного кодування даних ad-hoc мережі IoT на основі недетермінованої машини Тьюринга

Побудована у попередньому розділі 2.2 прикладна машина Тьюринга (АМТ) визначає загальний тип предметно-орієнтованої МТ. Для ідентифікації конкретного типу АМТ, замість літери «А» (Applied) застосуємо специфічну ознаку типу. Зокрема, визначимо тип «Недетермінована машина Тьюринга» (“Nondeterministic Turing Machine” (NdTM), в якій структурний компонент «FG» (граматика) є варіативною, наприклад, динамічно змінюваною у часі.

Ідея розробки спеціалізованої машини Тьюринга типу NdTM для крипто-стійкого динамічного кодування даних в цифровому каналі полягає у додатковому і незалежному крипто-стійкому кодуванні (пост-шифруванні) фреймів канального рівня поряд з відомими методами і протоколами шифрування даних. Завдяки цьому, здійснюється непередбачуване для третіх сторін «заплутування» структури фреймів, що ускладнює або практично унеможливлює оперативне перехоплення і аналіз змістовної інформації в потоці даних.

Сформулюємо принципи крипто-стійкого динамічного кодування фреймів канального рівня з урахуванням конструкції прикладної машини Тьюринга (АМТ, розділ 2.2) відповідно до структури інтерфейсів роботизованого мобільного комплексу Інтернету речей RIF (розділ 2.1, рис. 2.3). В основу алгоритму покладемо *принцип випадкового перемішування трійкових кодів для тегів (команд) розмітки цифрового потоку*.

Система команд АМТ (див. рис. 2.6) містить 14 строчок, кожна з яких визначає трійковий код і семантичний зміст відповідної команди як тегу розмітки. Будь яка перестановка місцями номерів і кодів в перших двох колонках системи команд АМТ змінює трійкові коди тегів і способи виконання відповідних команд машиною Тьюринга).

Кількість можливих перестановок P(N) на множині з N=14 елементів обчислюється за відомою формулою комбінаторики [131]:

$$P(N) = P(14) = 14! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot \ldots \cdot 14 = 87’178’291’200 \text{ (37 біт)}.$$

Довільну комбінацію з 14 різних натуральних чисел (кожне в діапазоні від 1 до 14) представимо як кортеж **k** з 14 блоків по 4 біти (56 біт); даний кортеж з 56 біт розмістимо у 7 блоків по 8 біт (тобто у 7 байт).

Кортеж **k** визначає спосіб перемішування 14 різних чисел, тобто «шаблон» (pattern - pat) що відображує лінійно впорядковану множину з 14 натуральних чисел [1, 2, 3, 4, ..., 14] у деяку невпорядковану множину з тих самих 14 чисел, наприклад,

$$\mathbf{k}_{pat1} := [1, 2, 3, 4, ..., 14] \rightarrow [3, 5, 14, 6, 1, 2, 7, 12, 4, 8, 10, 9, 11, 13].$$

Множина кортежів K:={k} містить P(14) = 14! = 87’178’291’200 екземплярів $k \in K$. Множину кортежів K:={k} назвемо *системним ключом кодування* (System Encryption Key - SEK) машини Тьюринга NdTM.

Нехай кортеж $k_t \in K$ визначає поточний стан S(t) недетермінованої машини Тьюринга як типу скінченного автомату Мілі. Назвемо кортеж $\mathbf{k_t} \in K$ *поточним ключом* крипто-стійкого кодування цифрового потоку (Current Encryption Key - CEK).

Визначимо *тимчасовий ключ кодування* (Temporary Encryption Key -TEK) $\mathbf{k_n} \in K_n \subset K$ для блоків даних у цифровому потоці в якості шаблону Pat для перемішування окремих біт у довільному блоці даних розміром ≥ 4 біт (напр., $\mathbf{k}_{pat1}$ [0 1 1 0 1 1] → [1 1 1 0 1 0]). Визначимо функцію V(n) від натурального аргументу n для крипто-кодування заголовку фрейму

$$V(n) := [K_n, H_n], n \in [1, 14!], \text{ де} \quad (1)$$

$K_n = \{\mathbf{k_n}\} \subset K$ – набір тимчасових ключів кодування (TEK); $H_n = \{(S, R)_n\}$ – набір векторів ідентичності відправника S та отримувача R, закодований ключом $\mathbf{k_n}$.

Побудуємо об’єкт “Frame Parameter Code” (FPC) для крипто-кодування даних на підрівні LLC (Logical Link Control) як кортеж даних наступного виду:

$$FPC := [n, H, B], n \in [1, 2, 3, ..., N], \text{ де} \quad (2)$$

n – параметр з формули (1), що використовується як «номер фрейму» і визначає *тимчасовий ключ* крипто-кодування $\mathbf{k_n}$;

$H_n = (S, R)_n$ – прихований заголовок фрейму, параметри якого закодовані *тимчасовим ключом* крипто-кодування $\mathbf{k_n}$ у формулі (1);

B_n – «тіло» фрейму (фрейм без заголовку та контрольної суми) що закодовано *тимчасовим ключом* крипто-кодування $\mathbf{k_n}$ у формулі (1).

Для передачі об’єкту FPC на підрівні MAC, він перетворюється у послідовність символів шляхом крипто-кодуванням тегів розмітки *поточним ключом* $\mathbf{k_t}$, а також доповнюється службовими структурними елементами:

- між-фреймовий захисний інтервал IFG (Inter Frame Gap);
- преамбула синхронізації (PR);
- делімітер фрейму SF (Start Frame delimiter);
- контрольна сума об’єкту FPC (FC) з формули (2).

Визначимо об’єкт підрівня MAC, який назвемо «динамічний код фрейму» (“Frame Dynamic Code” - FDC), у вигляді

$$FDC= \{[PR], [SF], [FPC], [FC]\}, \text{ де} \qquad (3)$$

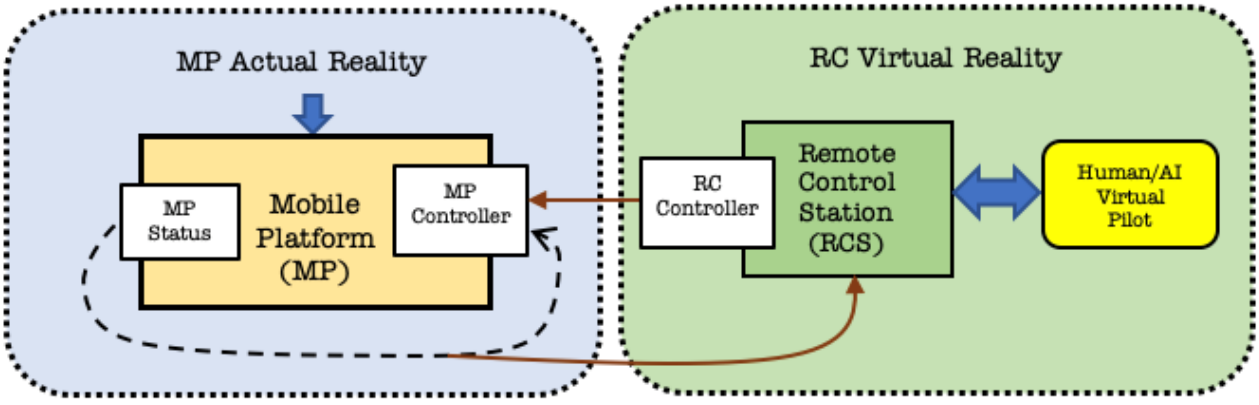
PR – преамбула для фізичної синхронізації детектора приймача з вхідним потоком лінійних сигналів, якими закодовано трійкові символи алфавіту MT;

SF – делімітер для логічної синхронізації цифрового потоку (визначення позиції першого трійкового символу фрейму як такого (тобто, об’єкту FPC);

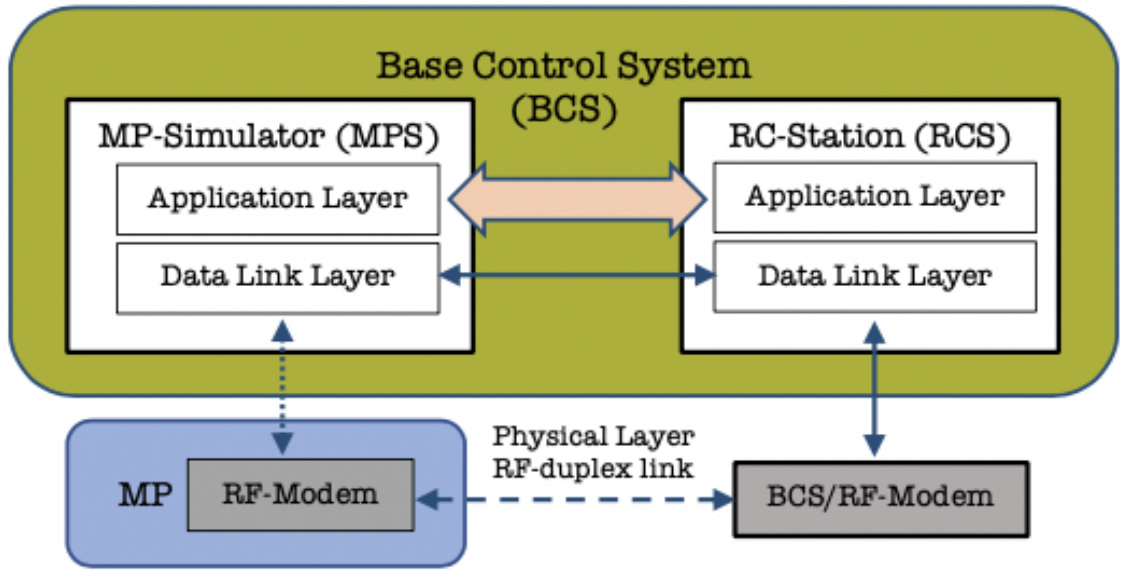
FPC - об’єкт змістовних параметрів фрейму (2);

FC - контрольна сума об’єкту FPC.

Розглянемо передумови недетермінованої машини Тьюринга NdTM для динамічного крипто-стійкого кодування на прикладі взаємодії станції дистанційного управління RCS з мобільною платформою MP, як показано нижче (див. оригінал на рис 2.2).



У свою чергу, станція дистанційного управління RCS входить до складу базової системи управління BCS, як показано нижче (див. оригінал на рис. 2.3).



Блок управління RC станції RCS підключено до радіо-модему базової системи управління (BCS/RF-Modem), через який здійснюється взаємодія з RF-модемом мобільної платформи MP на фізичному рівні лінійно-кодових сигналів трійкового алфавіту машини Тьюринга («_», «0», «1»).

Передумови двосторонньої взаємодії за алгоритмом NdTM.

Базова система управління BCS та мобільна платформа MP «знають» MAC-адреси інтерфейсів одне одного та своїх партнерів. Перед початком місії M, система BCS генерує унікальну для даної місії випадкову або псевдо-випадкову послідовність кортежів $K_M:=\{k_i\}$, де $k_i \in K=\{k\}$ належить множині K системного ключа SEK для крипто-стійкого динамічного кодування.

Послідовність $K_M=\{k_i\}$ назвемо *системним ключом місії* (Mission System Key - MSK). На відміну від системного ключа *кодування* SEK що визначено вище як $K:=\{k\}$, MSK є непередбачуваним, унікальним і одноразовим для кожної окремої місії, а також може мати різний розмір (тобто, кількість кортежів k_i), який залежить від тривалості місії і необхідного рівня захисту.

Окрім системного ключа місії MSK, перед початком місії ще генерується випадкове початкове значення поточного ключа $k_{t=0}$ місії CEK (Current Encryption Key), де $k_{t=0} \in$MSK крипто-стійкого кодування $k_t \in$ MSK.

В процесі виконання місії, значення поточного ключа CEK динамічно змінюється взаємодіючими сторонами за спеціальною процедурою взаємної синхронізації. Поточний ключ місії CEK відповідає за перемішування тегів розмітки цифрового потоку.

Безпосередньо перед стартом місії (за спеціальною процедурою, яка виключає вплив людського фактору), ідентичні копії ключа MSK і початкового поточного ключа CEK вводяться в пам'ять контролера RC станції дистанційного управління RCS та контролерів MC кожної мобільної платформи MP. При цьому, жодна людина, у тому числі, оператор базової системи управління BCS, не може дізнатися чи скопіювати системний ключ місії MSK, який відіграє роль головного «секрету» місії.

Натомість, «секрет» MSK є одноразовим, незалежно від того, чи мобільна платформа є одноразовою (типу «камікадзе») чи багаторазовою, а «секрет» поточного ключа CEK динамічно змінюється у часі, що унеможливлює накопичення необхідної статистики використання цього ключа з метою його зламу.

3) Базовий «Профіль_1» машини Тьюринга типу NdTM.

Після старту місії мобільні платформи MP і станція дистанційного управління RCS використовують ідентичні копії системного ключа місії MSK. За спеціальною процедурою, ключ MSK динамічно змінюються, а поточний ключ крипто-стійкого кодування CEK (кортеж $\mathbf{k_t} \in$ K) синхронізується. Ключом $\mathbf{k_t}$ здійснюється пост-шифрування цифрового потоку.

Ключ CEK визначає фізичні коди для тегів розмітки потока, а випадкове перемішування тегів заплутує логічну структуру фреймів та/або його окремих складових. При цьому, ключ CEK не впливає на числові параметри потоку (корисні та службові дані).

4) Спеціальний «Профіль_2» машини Тьюринга типу NdTM

Окрім систематичного використання динамічно-змінюваного ключа CEK, епізодично-випадково (в залежності від поточної ситуації) застосовується додаткове пост-шифрування параметрів потоку (корисних і службових даних) за допомогою тимчасового ключа TEK (Temporary Encryption Key). Ключ TEK передається відправником через «умовно-відкритий» випадковий параметр n (мітка фрейма, в якому корисні та/або службові дані додатково зашифровані ключом TEK). Параметр n визначає кортеж $\mathbf{k_n}=K_M(n)$ у послідовності системного ключа MSK (ідентичні копії K_M ключа MSK перед початком місії генеруються системою BCS і заносяться в пам'ять контролера RC станції дистанційного управління RCS та контролерів MC мобільних платформ MP). Додаткове пост-шифрування службових параметрів і корисних даних потоку ключем TEK ініціює відправник.

5) Спеціальний «Профіль_3» машини Тьюринга типу NdTM.

Обмін даними здійснюється за процедурою автентифікації сторін з використанням тимчасового ключа TEK як кортежу $K_M(n)=\mathbf{k_n}$. Передавач, за формулою (1) на послідовності K_M замість K, реалізує функцію $V(n):=[K_M, H)]_n$, де N – довжина послідовності K_M; $H_n=(S, R)_n$ – закодований ключом $\mathbf{k_n}$ заголовок (вектор-ідентичності MAC-адрес відправника S і отримувача R). При цьому корисні дані необов'язково шифруються ключем TEK.

6) Спеціальний «Профіль_4» машини Тьюринга типу NdTM.

Високо-захищена взаємодія з використанням процедури автентифікації сторін і додатковим пост-шифруванням даних тимчасовим ключем TEK.

Відправник S, за формулою (2), створює об'єкт FPC:= [n, H, B], $n\in[1, 2, 3, \ldots, N]$ з довільним випадковим номером «n», де заголовок H (вектор-ідентичності відправника S і отримувача R) та тіло B зашифровані ключем $\mathbf{k_n}=K_M(n)$ відповідно до номеру «n» об'єкта FPC і власної копії ключа K_M. Далі, об'єкт FPC крипто-кодується поточним ключем $\mathbf{k_t}$ місії (CEK) у послідовність лінійних символів для передачі інформації на фізичному рівні каналу зв'язку.

Одержувач R, у зворотному порядку, дешифрує символи лінійного коду ключем CEK і перетворює їх в об'єкт FPC, потім декодує заголовок H ключем $\mathbf{k_n}$=K(n) відповідно до номеру «n» об'єкта FPC і власної копії ключа K_M. Якщо ідентифікатори (MAC-адреси) відправника та одержувача є коректними (входять до наперед узгодженого списку взаємодіючих сторін), то тіло кадру далі розшифровується та обробляється; інакше об'єкт FPC відкидається.

Розглянуті вище способи формування об'єктів і правила визначають конкретний «скінчений автомат» FSM (Finite State Machine) у вигляді прикладної недетермінованої машини Тьюринга NdTM для крипто-стійкого динамічного кодування даних в мережах Інтернету речей:

- FPC (Frame Parameter Code) – структурно-параметричний код фрейму;
- FDC (Frame Dynamic Code) – динамічний код фрейму на підрівні MAC;
- функція V(n) прихованого кодування заголовку фрейму – формула (1);
- 8 правил синтаксису формальної граматики FG;
- 3 правила семантики (алгоритм) захищеної взаємодії відправника S і одержувача R;
- система 14-ти команд потокового кодування (див. рис.2.6).

Далі розглядаються деякі типові профілі машини Тьюринга NdTM.

1) Оперативна 3-канальна взаємодія в реальному часі (Профіль-1v1)

$$\begin{cases} \mathbf{\{...\}} := \mathbf{\{TMF}, \mathbf{TMF} \ldots \mathbf{\}}; \\ \mathbf{TMF} = \mathbf{[B\ (}\mathrm{RTD}\mathbf{,}\ \mathrm{RTD}\mathbf{,}\ \mathrm{RTD}\mathbf{),\ FC]} \\ \mathrm{RTD} = \text{“bbbb...b”} \end{cases} \quad (4)$$

Даний профіль передбачає відомі сторонам адреси відправника (базової станції BS) та одержувача (мобільного об'єкта MO) і незмінні протягом однієї сесії. Телеметрія даних реального часу (RTD) передається короткими фреймами MT (TMF). Кожен TMF містить тіло B з декількох (в даному випадку - трьох) сегментів RTD розміром від 4 біт, а також контрольну суму FC). Автентифікація відсутня. Мінімізується надлишковість і збільшується інформаційна ефективність каналу, що важливо при обміні короткими даними телеметрії в умовах завад чи високому рівні помилок BER (Bit Error Rate).

Блоки RTD у фреймі TMF визначають кількість каналів від 0 (пустий фрейм) до декількох в залежності від пропускної спроможності каналу.

Додатковий крипто-захист (пост-шифрування) каналу взаємодії реалізується тим, що фізичні коди тегів розмітки недетермінованої машини Тьюринга на підрівні MAC відомі лише взаємодіючим сторонам через їхні власні ідентичні копії унікального одноразового системного ключа місії MSK у вигляді випадкової послідовності кортежів K_M, а також через поточне значення $\mathbf{k_t} \in K_M$ цього ключа – Current Encryption Key (CEK) що динамічно змінюється протягом всього часу виконання місії.

Максимальний розмір ключа K_M (без повторів кортежей) дорівнює кількості кортежей системного ключа SEK (System Encryption Key) – множині K перестановок в наборі з 14 кодів команд прикладної машини Тьюринга AMT: P(14)=14! = 87’178’291’200. Ключ K_M випадковими кортежами з множини K, і завантажується базовою станцією BCS безпосередньо в мобільну платформу MP перед початком кожної окремої місії за спеціальною процедурою, яка практично виключає вплив людського фактору. Жодна людина (у тому числі, оператор базової станції) не має доступу до ключа K_M.

2) Багатоканальна передача даних з автентифікацією (Профіль-3v1)

Розглянемо формалізований запис одного сеансу три-канальної передачі даних за процедурою взаємної автентифікації сторін:

$$\begin{cases} \{...\} := \{\mathbf{TMF}_0\ ,\ \mathbf{TMF}_1\ ,\ ....\ ,\ \mathbf{TMF}_3\}; \\ \mathbf{TMF}_0 = [\ (01101011)\ ,\ \mathbf{H}\ (00011101,\ 01111100)\ ,\ \mathbf{FC}\ ]; \\ \mathbf{TMF}_m = [\ \mathbf{B_m}\ (bbbb...b)\ ,\ \mathbf{FC_m}\ ],\ m=1,\ 2,\ 3. \end{cases} \qquad (5)$$

Сеанс (5) складається з чотирьох фреймів. Фрейм TMF_0 містить випадковий номер сеансу **n** (перший параметр «01101011»), а також MAC-адреси відправника S та одержувача R (два параметри заголовку H), які зашифровані тимчасовим ключом крипто-кодування $\mathbf{k_n}$ за формулою (1). Ідентифікатор відправника S_{ID} є перший параметр заголовку H у фреймі $\mathbf{TMF}_0$ (крипто-кодована MAC-адреса “00011101”).

Ідентифікатор одержувача R_{ID} є другий параметр заголовку H (крипто-кодована MAC-адреса “01111100”). Фрейми $TMF_1 \div TMF_3$ містять блоки корисних даних $\mathbf{B_m}$ **(**bbbb...b**)** і контрольні суми кожного фрейму $\mathbf{FC_m}$.

Обидві сторони (відправник S та одержувач R), використовують ідентичні копії системного ключа місії MSK, а також поточний ключ крипто-стійкого кодування CEK (кортеж $\mathbf{k_t} \in K$) що динамічно змінюється і взаємно синхронізується за спеціальною процедурою. Кортежом $\mathbf{k_t}$ поточного ключа крипто-стійкого кодування CEK здійснюється пост-шифрування тегів розмітки цифрового потоку, яке визначає динамічно-змінні фізичні коди для тегів розмітки потока (випадкове перемішування цих тегів заплутує логічну структуру фреймів і ускладнює їх перехоплення). При цьому, ключ CEK не впливає на числові параметри потоку (корисні та службові дані).

Одержувач R декодує фрейм TMF_0 ключом $\mathbf{k_t}$, визначає кортеж $\mathbf{k_n}$ за номером **n** (тимчасовий ключ TEK), яким дешифрує ідентифікатори S_{ID}, R_{ID}; якщо вони збігаються з попередньо записаними в пам’яті MAC-адресами сторін, то корисні дані сприймаються, інакше - ігноруються. Додаткове пост-шифрування корисних тимчасовим даних ключом $\mathbf{k_n}$ не застосовується.

Висновки до розділу 2

В даному розділі дисертаційної роботи обґрунтовано основні теоретичні положення щодо синтезу моделі і методу крипто-стійкого динамічного кодування даних для підвищення рівня кіберзахисту мобільної ad-hoc мережі Інтернету речей. Сформульовано перші три наукові результати роботи.

Досліджено еволюцію екосистеми робототехніки “Robotics Eco-System”, а також методи і моделі автоматизованого проектування роботизованих мобільних комплексів і платформ. Проаналізовано особливості кодування і структуризації даних у технологіях локальних мереж Ethernet та її бездротового аналогу технології WiFi. Обґрунтовано багаторівнева формальна граматика для гнучкої структуризації фреймів мобільної ad-hoc мережі Інтернету речей.

Розглянуто узагальнену функціональну схему дистанційного управління мобільною платформою Інтернету речей. Розроблено трирівневу структуру інтерфейсів роботизованого мобільного комплексу, а також формальну граматику побудови фреймів в каналі зв’язку з трійковим лінійними сигналами для нестандартного утворення цифрового потоку з окремих незалежних функціональних елементів у довільному порядку і складі. На основі цього, удосконалено модель динамічного кодування даних мобільної ad-hoc мережі Інтернету речей, що дозволяє збільшити ефективність каналу у 1.5÷6.6 разів та ускладнити його злам за рахунок скорочення службової інформації і непередбачуваного формату даних (**перший науковий результат**).

Визначено систему команд і узагальнену формальну граматику для прикладної машини Тьюринга з правилами семантики потокового кодування фреймів. Розроблено недетерміновану машину Тьюринга для структуризації даних в каналі зв’язку, відмінну тим, що через випадкові інтервали часу, кодові кортежі команд перемішуються і змінюють свою семантику у порядку відомому лише конкретному адресату. Таким чином, отримала подальший розвиток методика застосування теорії автоматів, що дозволяє створювати різноманіття алгоритмів криптографічного захисту інформації незалежно від існуючих апаратно-програмних засобів (**другий науковий результат**).

Вперше запропоновано спосіб динамічної синхронізації профільного ключа крипто-стійкого кодування між бортовим контролером мобільного об’єкту та базовою станцією дистанційного управління на основі недетермінованої машини Тьюринга, відмінний тим, що на кожну місію мобільного об’єкту генерується і закладається в пам’ять обох сторін унікальна випадкова послідовність профільних ключів, а номер ключа у цій послідовності відіграє роль відкритого ключа для передачі по каналу зв’язку, що дозволяє підвищити проактивний кіберзахист мережі, спростити його апаратно-програмну реалізацію і підвищити ефективність каналів зв’язку, а також обмежити людський фактор у питаннях конфіденційності (**третій науковий результат**).

РОЗДІЛ 3
РОЗРОБКА КОМП'ЮТЕРНИХ АЛГОРИТМІВ КРИПТО-СТІЙКОГО ДИНАМІЧНОГО КОДУВАННЯ І РОЗПОДІЛУ ДАНИХ AD-HOC МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ

В даному розділі розробляються і досліджуються два комп'ютерні алгоритми відповідно для крипто-стійкого динамічного кодування та розподілу даних мобільної ad-hoc мережі Інтернету речей (IoT).

Перший алгоритм призначається для крипто-стійкого динамічного кодування даних на недетермінованій машині Тьюринга, теоретичні засади якого сформульовані у розділі 2 даної роботи. Визначаються базові профіля алгоритму. Будується блок-схема і програма. Проводиться оцінка ефективності.

Другий алгоритм призначається для динамічного розподілу даних в мережі IoT на основі комбінаторної моделі оптимізації потоків з використанням штучного інтелекту, яка формалізується у підрозділі 3.2 даної роботи.

Досліджуються характеристики нормалізованого планарного графу ad-hoc мережі. Розширюються базові формули комбінаторики для обчислення параметрів нормалізованого графу. Аналізується топологічна структура шляхів ST-графу з точки зору побудови дерева пошуку найкоротших шляхів.

Обґрунтовується комбінаторна модель оптимального розподілу даних в ad-hoc мережі IoT. Надається методика рекурсивної побудови дерева пошуку найкоротших шляхів ST-графу за допомогою штучного інтелекту. Виводиться формула обчислювальної складності для алгоритму оптимального розподілу даних по шляхах ST-графу.

Імплементація даного алгоритму спрямовується на підвищення загальної функціональної стійкості мобільних ad-hoc мереж за рахунок оптимізації її структури і параметрів в реальному часі в умовах зовнішніх впливів та активної протидії.

3.1. Комп'ютерний алгоритм крипто-стійкого динамічного кодування в ad-hoc мережі IoT на основі недетермінованої машини Тьюринга

В даному підрозділі розробляється і досліджується алгоритм крипто-стійкого динамічного кодування даних в мобільній ad-hoc мережі Інтернету речей (IoT) на недетермінованій машині Тьюринга (NdTM), теоретичні засади якого сформульовані вище у розділі 2 даної роботи.

Визначимо наступні базові профілі роботи алгоритму.

Профіль 1. Гнучке формування нестандартних фреймів тегами розмітки цифрового потоку за умови відсутності кіберзагроз і в залежності від характеру корисних даних та стану каналу зв'язку. Даний профіль спрямовується на *підвищення інформаційної ефективності* каналу зв'язку за рахунок зменшення службової інформації по відношенню до корисного навантаження фрейму.

Профіль 2. Крипто-стійке кодування фреймів поточним ключом кодування CEK (Current Encryption Key, $\mathbf{k_t}$) без використання відомих апаратно-програмних методів шифрування.

Профіль 3. Додаткове крипто-стійке кодування (пост-шифрування) корисних і службових даних тимчасовим ключом кодування TEK (Temporary Encryption Key, $\mathbf{k_n}$) з автентифікації сторін, без використання відомих методів шифрування. Забезпечує *підвищений захист простими* засобами.

CEK – це елемент $\mathbf{k_t} \in K_M := \{k_t\} \subset K$ системного ключа місії K_M (Mission System Key, MSK) з масиву кортежів $\{k\}$, кожен з яких є деякою перестановкою цілих чисел від 1 до 14 (команд машини Тьюринга NdTM). У свою чергу, системний ключ місії MSK (масив K_M) є підмножиною системного ключа кодування SEK (System Encryption Key) машини Тьюринга (NdTM): $K_M \subset K$.

TEK – це елемент $\mathbf{k_n} \in K_N := \{k_n\}\} \subset K$ з набору тимчасових ключів кодування K_N. Системний ключ місії K_M та набор тимчасових ключів кодування K_N випадково генеруються і завантажується в пам'ять об'єктів-учасників місії перед початком її виконання (див. роз. 2.3).

Функціональна схема *кодування вихідних даних* на передавальній стороні каналу зв'язку IoT за допомогою недетермінованої машини Тьюринга NdTM показана на рисунку 14. Модуль Input Sequential Text (IST) імітує вхідний пакет завдання на «мові програмування FG» – формальній граматиці машини Тьюринга (три правила потокового кодування фреймів FSC з 14-ма командами1, див. рис.13). Генераторами вхідних пакетів IST можуть бути людина чи кібер-оператор станції дистанційного управління RCS (див. рис. 2.3), спеціалізований бортовий контролер мобільної платформи MP, тощо.

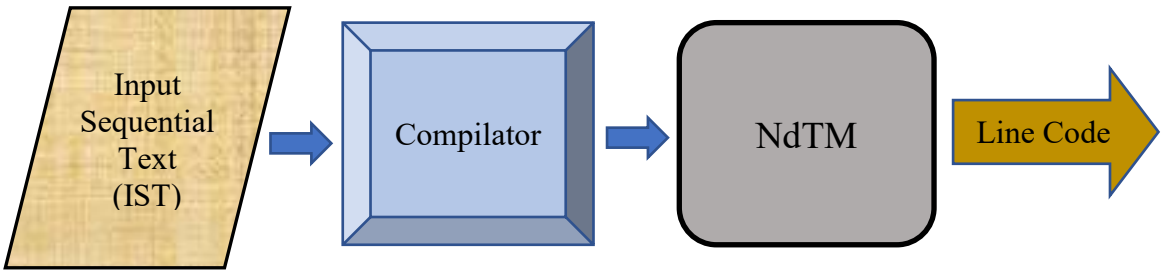


Рисунок 3.1 – Функціональна схема кодування даних в каналі зв'язку за допомогою недетермінованої машини Тьюринга NdTM

Модуль «Compilator» аналізує завдання IST на відповідність правилам формальної граматики FG. При відсутності помилок, послідовність команд і даних вхідного пакету передається на вхід модуля NdTM (інакше компілятор генерує зворотне діагностичне повідомлення про виявлені помилки тексту IST).

Модуль NdTM імітує процесор 14-ма командами (див. рис. 2.6). Перші 13 команд є тегами розмітки потоку символів на канальному рівні (їхні бінарні коди мають не більше 3 біт). Псевдо-команда 14 розширює систему команд NdTM службовими командами від 4 і більше біт, за допомогою яких контролюється внутрішній стан NdTM, зокрема, начальне завантаження системного ключа крипто-кодування SEK (множини кортежів місії K_M), установка і динамічна зміна поточного ключа кодування $\mathbf{k_t} \in K_M$, активація тимчасовго ключа шифрування TEK (кортежу $\mathbf{k_n} \in K_M$) тощо. На виході, NdTM генерує трійковий лінійний код тегів розмітки і корисних/службових даних.

Функціональна схема зворотного процесу *декодування вхідних даних* на приймальній стороні каналу зв'язку IoT за допомогою недетермінованої машини Тьюринга NdTM показана на рисунку 3.2.

Лінійний код (Line Code, LC), як послідовність трійкових фізичних сигналів для тегів розмітки потоку, службових команд, а також корисних даних і службових параметрів, поступає на вхід модуля NdTM приймальної сторони окремими сеансами двостороннього зв'язку в межах однієї чи декількох сесій конкретної місії.

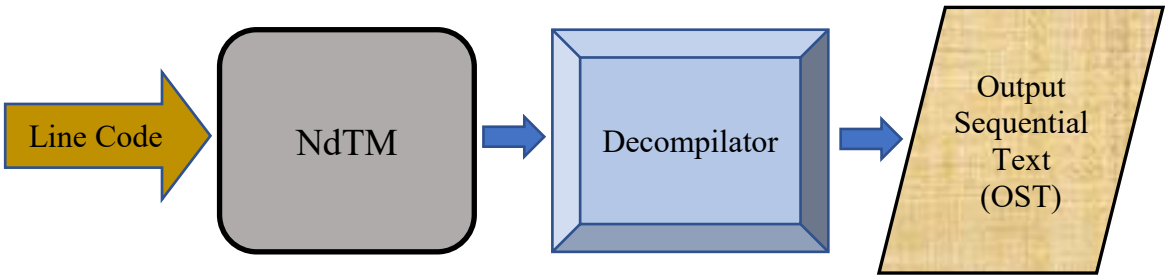


Рисунок 3.2 – Функціональна схема декодування даних в каналі зв'язку за допомогою недетермінованої машини Тьюринга NdTM

Лінійний код LC в цілому за умовчанням крипто-кодується поточним значенням $\mathbf{k_t}$ системного ключа кодування SEK. Натомість, корисні дані та службові параметри лінійного коду LC, – в залежності від профіля і версії роботи відповідного модуля NdTM на передавальній стороні, – можуть бути:

1) нешифрованими;

2) шифрованими тільки додатковим ключем пост-шифрованими $\mathbf{k_n}$;

3) шифрованими тільки одним з відомих методів (напр., WPA2);

4) шифрованими відомим методом та ключем пост-шифрування $\mathbf{k_n}$.

Приймальний модуль NdTM декодує фізичні сигнали лінійного коду LC у послідовність символів формальної граматики FG машини Тьюринга поточним значенням $\mathbf{k_t}$ системного ключа SEK. Декомпілятор перевіряє коректність тексту, дешифрує (при необхідності) корисні дані та службові параметри ключем $\mathbf{k_n}$, і формує модуль вихідних даних OST. При цьому може знадобитись подальше дешифрування OST відомим методом (напр., WPA2).

Програмний алгоритм прийому-передачі крипто-кодованих даних за допомогою недетермінованої машини Тьюринга NdTM реалізуємо у середовищі MAC/OS на мові програмування Python модулем TMsend (Turing Machine Send) відповідно до функціональної схеми, зображеної вище на рисунку 3.1.

1) Представимо команди машини Тьюринга (див. рис. 2.6) як двомірний масив символів TMcom(2, 15), де значення індексів «0» є умовними для відповідності номерам команд. Бінарні коди команд доповнимо знаком «пробіл» («_») з алфавіту машини Тьюринга як показано нижче на рисунку 3.3.

TMcom = **[**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	
.	**,**	**;**	**:**	**(**	**)**	**H**	**B**	**FC**	**[**	**]**	**{**	**}**	**$**	],
_0	**_1**	**_00**	**_01**	**_10**	**_11**	**_000**	**_001**	**_010**	**_011**	**_100**	**_101**	**_110**	**_111**	],

]

№	Код	Мнемо	Семантика	Semantics
1	0	.	Точка	Dot
2	1	,	Кома	Comma
3	00	;	Крапка з комою	Semicolon
4	01	:	Двокрапка	Colon
5	10	(	Ліва дужка	Left Parenthesis
6	11	)	Права дужка	Right Parenthesis
7	000	H	Заголовок	Header
8	001	B	Тіло	Body
9	010	FC	Контрольна сума	Frame Check sequence
10	011	[	Ліва скобка	Left Bracket
11	100	]	Права скобка	Right Bracket
12	101	{	Ліва фігурна скобка	Left Brace
13	110	}	Права фігурна скобка	Right Brace
14	111	**$**	Розширення команд ≥ 4 біт)	Extension

№	0	1	2	3	4	5	6	7
Symbol	_	**.**	**,**	**;**	**:**	**(**	**)**	**H**
Code	_	**_0**	**_1**	**_00**	**_01**	**_10**	**_11**	**_000**

№	0	8	9	10	11	12	13	14
Symbol	_	**B**	**FC**	**[**	**]**	**{**	**}**	**$**
Code	_	**_001**	**_010**	**_011**	**_100**	**_101**	**_110**	**_111**

Рисунок 3.3 – Відображення команд NdTM у трійкові символи лінійного коду

2) Системний ключ кодування SEK представимо як двомірний масив цілих чисел (K, 15); K – кількість кортежів системного ключа; (15-1) – кількість команд МТ, що перемішані в кортежах $\mathbf{k_t}$ системного ключа SEK. Для K = 3:

SEK = [
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14],
[0, **3**, 2, **1**, 4, **9**, 6, 7, 8, **5**, 10, **13**, 12, **11**, 14],
[0, 1, 2, 3, **7**, 5, 6, **4**, 8, 9, **13**, **14**, 12, **10**, **11**]
]

3) Кортеж поточного системного ключа $\mathbf{k_t}$ в масиві кортежів SEK позначимо цілою змінною Kt, що за умовчанням дорівнює Kt=SEK(**Kt_index**); при старті програми Kt індекс можна змінювати від 0 до K–1.

4) Пакет вхідних даних (див. рис. 14) задекларуємо як одномірний масив символів **InPack(N)** розміром N. Наприклад, у послідовності за формулою (5),

{...}:= **{TMF**$_0$, **TMF**$_1$ **, ... , TMF**$_3$**}**;
TMF$_0$ = **[(**01101011**) , H (**00011101**,** 01111100**) , FC]**;
TMF$_m$ = **[B**$_m$ **(**bbbb...b**) , FC**$_m$ **]**, m=1, 2, 3

блок **[(**01101011**)** пакету INPack матиме вигляд:

InPack = ['**[**', '**(**', '**0**', '**1**', '**1**', '**0**', '**1**', '**0**', '**1**', '**1**','**)**', '**,**'] .

Принцип дії програмного модуля TMSend. По значенню Kt_index, зчитується кортеж Kt системного ключа: Kt=SEK(Kt_index). Далі формуються 2 списки: а) кодова таблиця Code_tab(2, 15), де строка 0 – теги команд МТ, строка 1 – теги підстановки (кортеж Kt); б) таблиця InOut_tab (2, N), де строка 0 – вхідний пакет In_pack(N), строка 1 – вихідний пакет Out_pack(N) що спочатку копіює вхідний. В циклі по ind∈[1,N] зчитується черговий символ In_Smb = InOut(0, ind). У вкладеному циклі по cmd в строчці 0 таблиці Code_tab(0) знаходиться номер n_tag для символу In_Smb (якщо він є тегом). По номеру n_tag зчитується тег підстановки Sub_tag = Code_tab(1, n_tag). В строку 1 таблиці InOut_txt записується тег підстановки: InOut_pack(1, n_tag) = Sub_tag. Вхідні символи «0», «1» (дані, параметри) залишаються незмінними.

Блок-схема програмного модуля TMSend
для крипто-кодування даних недетермінованою
машиною Тьюринга NdTM

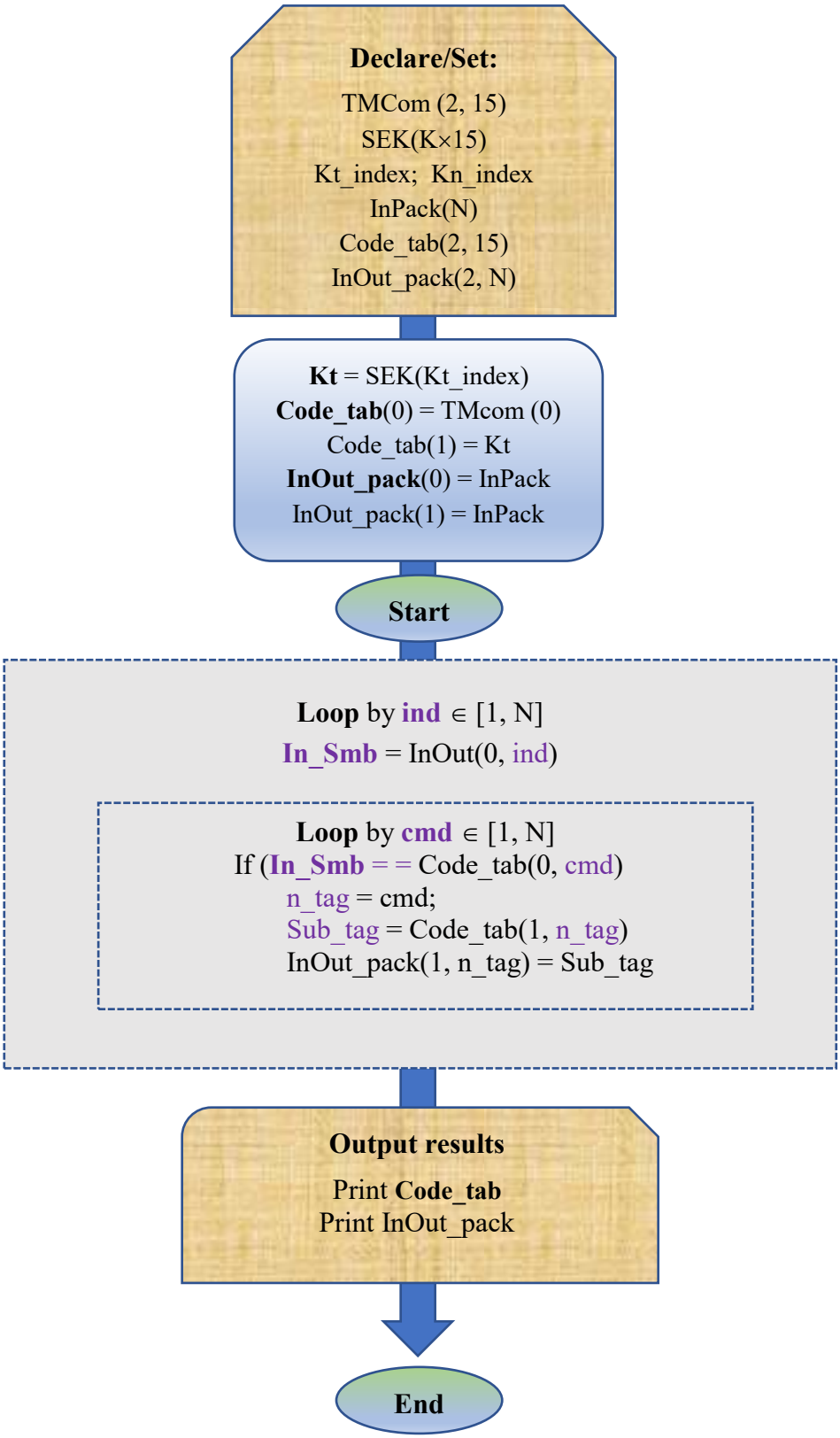


Рисунок 3.4 – Блок-схема програмного модуля TMSend

Структура вхідних параметрів програмного модуля TMSend

TMCom (2, 15) – двомірний масив 14 команд машини Тьюринга у форматі

№	0	1	...	14
Symbol	_	.		**$**
Code	_	**_0**		**_111**

де TMCom(1, 0) = '.'; TMCom(1, 1) = '_0'; TMCom(14, 0) = '$'; TMCom(14,1) = '_111'. TMCom(0, 0), TMCom(0,1) – незначущі елементи, заповнені «пробілами».

SEK(K×15) – двомірний масив з K кортежів системного ключа крипто-кодування у форматі

k	0	1	...	14
0	0	**1**	**...**	**14**
1	0	**3**	**...**	**14**
2	0	**1**	**...**	**11**

де SEK(0, 1÷14) – кортеж $\mathbf{k_t}(0)$ = '1. ..., 14'; SEK(1, 1÷14) = $\mathbf{k_t}(1)$ = '3. ..., 14';

SEK(2, 1÷14) = $\mathbf{k_t}(2)$ = '1. ..., 11'.

Kt_index – цілий стартовий індекс поточного системного ключа крипто-кодування $\mathbf{k_t}$, який визначає спосіб перемішування 14 номерів команд машини Тьюринга при кожному старті програми; за умовчанням, Kt_index =0. При старті програми, його можна змінювати від 0 до K–1. Поточне значення системного ключа крипто-кодування $\mathbf{k_t}$ симулюється в програмі змінним кортежом Kt = SEK(Kt_index).

Kn_index – цілий стартовий індекс поточного ключа пост-шифрування $\mathbf{k_n}$, який визначає спосіб додаткового шифрування корисних та службових даних при кожному старті програми. За умовчанням, Kn_index=0, що означає відсутність пост-шифрування; при старті його можна змінювати від 0 до K–1.

InPack(N) – пакет вхідних даних (строка символів).

Тестові приклади симуляції NdTM

Тест 1. Базове крипто-кодування

Системний ключ крипто-кодування SEK:

0. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
1. 0, **3**, 2, **1**, 4, **9**, 6, 7, 8, **5**, 10, **13**, 12, **11**, 14
2. 0, 1, 2, 3, **7**, 5, 6, **4**, 8, 9, **13**, **14**, 12, **10**, **11**

Kt_index = 0;

Індекс поточного ключа крипто-кодування:

Kt_index = 0 (базове значення);

Kt=SEK(**Kt_index**) = SEK(**0**) =

(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14);

Kn_index = 0; (пост-шифрування відсутнє);

Вхідний текст: IST = [(0 1 1 0 1 0 1 1)

У форматі Python: IST = ['**[**', '**(**', '**0**', '**1**', '**1**', '**0**', '**1**', '**0**', '**1**', '**1**','**)**', '**,**']

Результат крипто-кодування: (співпадає з IST)

OST = [(0 1 1 0 1 0 1 1)

Результат лінійного трійкового кодування:

OST = [(0 1 1 0 1 0 1 1)

OST3 = _011_10_0 1 1 0 1 0 1 1_11

Тест 2. Крипто-кодування довільним ключом Kt

Kt_index = 1; (альтернативний ключ)

Kt=SEK(Kt_index) = SEK(**1**) =

(0, **3**, 2, **1**, 4, **9**, 6, 7, 8, **5**, 10, **13**, 12, **11**, 14);

IST = [(0 1 1 0 1 0 1 1)

Результат крипто-кодування: (не співпадає з IST)

OST = [FC 0 1 1 0 1 0 1 1)

Результат лінійного трійкового кодування:

OST = [FC 0 1 1 0 1 0 1 1)

OST3 = _011_010_0 1 1 0 1 0 1 1_11 .

3.2. Комбінаторна модель оптимізації розподілу даних ad-hoc мережі Інтернету речей з використанням штучного інтелекту

В даному підрозділі дисертаційній роботі обґрунтовується удосконалений метод прискореного рішення комбінаторних задач на графах і відповідна комбінаторна модель оптимізації розподілу даних в ad-hoc мережі Інтернету речей (IoT) з використанням штучного інтелекту. Досліджуються властивості нормалізованого вільно-орієнтованого планарного графу ad-hoc мережі (ST-графу). Розширюються базові формули комбінаторики для обчислення параметрів ST-графу. Аналізується структура шляхів ST-графу для рекурсивної побудови дерева пошуку найкоротших шляхів ST-графу за допомогою ШІ. Виводиться формула обчислювальної складності для алгоритму оптимального розподілу даних по шляхах ST-графу.

Методи і алгоритми оптимізації потоків в комп'ютерних мережах і системах відіграють важливу роль у підвищенні продуктивності використання мережевих ресурсів в нестаціонарних умовах. Сучасні мережі IoT з оптичними і безпровідними лініями здатні динамічно налаштовувати свою структуру, зокрема, пропускну здатність каналів зв'язку у зустрічних напрямках.

Традиційне подання комп'ютерної мережі «логістичними» орієнтованими графами з фіксованою ємністю дуг в кожному напрямку обмежує потенційно можливі результати. За цих обставин, запропоновано альтернативну модель – «вільно-орієнтований зважений граф» (Free-oriented Weighted Graphs FWG) [132], 2020), а також метод обчислення максимального потоку на двополюсному планарному графі типу FWG ([133], 2023). Разом з тим, виявилася неточність оцінки складності методу через те, що не всі шляхи на графі є взаємно незалежними. Окрім того, метод лише знаходить максимальну величину загального потоку без його розподілу по шляхах ([13], 2025).

Задачі оптимізації на графах, зазвичай, мають комбінаторний характер, і ключовим аспектом є оцінка складності побудови «дерева пошуку» рішень, параметри якого визначають кількість ітерацій відповідних алгоритмів, необхідні ресурсі (час, пам'ять і швидкодію процесора) та вартість реалізації.

Частину з цих параметрів можна рахувати по відомих формулах комбінаторики та оптимізації, наприклад, максимальний потік по методам Форда-Фалкерсона, Дініца тощо. Кожен з них має свої сферу застосування і оцінки складності що визначається функцією O(V), де V – вектор властивостей графу, наприклад, кількість його вершин і ребер [64]-[70].

На рисунку 3.5 представлено розширене подання комбінаторних формул з урахуванням результатів комбінаторної оптимізації на нормалізованому ST-графі. Стилістичні правки базових формул стосуються рядків 2, 3, 4, 6 ([13]).

Формула 2 (поліноміальний коефіцієнт) є непростою для розуміння, однак, її легше усвідомити на прикладі знайомої плутанини у порядку літер при наборі окремих слів в редакторі Word. Якщо замість “mam” помилково введено “amm”, то є три очевидні альтернативи {amm, mam, mma}, з яких лише “mam” має коректний зміст і буде запропонована редактором Word для авто-підстановки. Тому замість P(n), введено ім’я W(n), яке асоціюється з редактором Word та написанням слів. Окрім того, для «множення за індексом», у формулі використано грецьку літеру П (аналогічно знаку Σ додавання за індексом).

Формула $N(n, k) = n^k$ у рядку 3 (Таб. 6) має типову наочну інтерпретацію – утворення різних цілих чисел впорядкованими наборами по k цифр в обраній системі числення з n цифр. Наприклад, двійкові числа (n=2, цифри: 0, 1); десяткові числа (n=10, цифри: 0, 1, 2, 3, ..., 9); шістнадцятирічні числа (n=16, цифри: 0, 1, 2, ..., 9, A, B, C, D, E, F). Легко бачити, що двома десятковими цифрами можна створити рівно $10^2 = 100$ різних чисел (від 0 до 99), трьома цифрами – рівно $10^3 = 1000$ чисел (від 0 до 999), і так далі. Англійською мовою «числа» - це “numbers’. За цієї логіки, замість традиційного ім’я A_n^k (т. зв. “Arrangements”, або «розміщення з повтореннями»), автори ввели ім’я “N”.

Формула 4, зазвичай, позначається як функція A_n^k , тобто повністю співпадає за своїм традиційним ім’ям з попередньою формулою 3. При цьому розуміння змісту кожної з формул 3 та 4 потребує додаткового роз’яснення (формула 3 – це «розміщення з повтореннями», а 4 – «розміщення без повторень»).

РОЗШИРЕНА ТАБЛИЦЯ КОМБІНАТОРИКИ				
Базові формули комбінаторики				
№	Позначення	Коментар (EN)	Коментар (UA)	Змістовна інтерпретація
1	**P(n) = n!**	**Permutations** of **n** elements	Перестановки (впорядковані множини з **n** елементів)	Розміщення **n** пасажирів в автобусі що має **n** місць Для пасажирів {a, b, c}: **n** = 3; **P(n)** = 3! = 1· 2 · 3 =**6**. Множина з 6 перестановок по місцях {1_, 2_, 3_}: {(a, b, c); (a, c, b); (b, a, c); (b, c, a); (c, a, b); (c, b, a)}
2	**W(n) = n! / Π(k_i!)**	**Words** of **n**-letter repeated	Слово-утворення з **n** літер з повторами	Набор літер {mam}: n =3 ; k_1(m) = 2; k_2(a) = 1. **W(n)** = 3! / (2!×1!) = **3**. Набор слів = {**amm, mam, mma**}
3	**N(n, k) = n^k**	**Numbers** by **k** of **n** elements	Число-утворення (впорядковані набори по **k** із **n** елементів з повторами)	Множина {x} **k**-розрядних чисел, утворених на множині з **n** цифрових символів. Для **k** = 3; **n** = 2: **N(n, k)** = n^k = 2^3 = **8**; {x}= {000, 001, 010, 011, 100, 101, 110, 111}
4	**A(n, k) = n!/(n–k)!**	**Arrangements** by **k** ≤ **n** of **n** elements	Угрупування (впорядковані вибірки по **k** із **n** елементів без повторів)	Команда **k** учасників змагань (з рейтингами) обраних із тренувальної группы **n** спортсменів. Для k = 2, n = 4 спортсмени {a, b, c, d}: **A(n, k)** = 4!/(4–2)! =4! / 2! = 3·4 = **12**. Можливі 12 варіантів складу команди: {ab, ac, ba, bc, ca, cb, ad, da, cd, dc, db, bd}; (ab≠ba)
5	**C(n, k) =** **n! /(k!· (n-k)!)**	**Combinations** by **k** ≤ **n** of **n** elements (binomial coefficients)	Поєднання: невпорядковані вибірки по **k** із **n** елементів без повторів (біномні коефіцієнти)	Команда **k** учасників змагань (без рейтингів) обраних із тренувальної группы **n** спортсменів. Для k = 2, n = 4 спортсмени {a, b, c, d}: **C(n, k)** = 4! / [2!· (4-2)!] = 4! / (2!·2!) = **6**. Можливі **6** варіантів складу команди: {(a, b); (a, c); (a, d); (b, c); (b, d); (c, d)}
6	**Ch(n, k) = (n+k-1)! / (k!· (n-1)!)**	**Choices** **k** things of **n** types	Вибірки: невпорядковані набори по **k** з **n** різних типів елементів	Крамниця має 3 типи цукерок. Купити 2 цукерки. n = 3; k = 2. **Ch(n, k)** = 4! / (2!×2!)= 2×3 = **6**. **{(a, a), (b, b), (c, c), {(a, b), (a, c), (b, c)}**
Формули комбінаторики для нормалізованого ST- планарного графу				
7	**GE(V) = V (V-1) / 2** **V ≥ 3**	**Graph Edges** on free-oriented full **V**-Graph	Кількість ребер у повному вільно-орієнтованому графі з **V** вершинами	Вільно-орієнтований зважений повний граф з **V** = 6 вершинами: **V**-Граф з кількістю вершин **V** = 6: **GE(V)** = 6 (6-1)/2 = **15**
8	**PGE (V) = 3 (V-2)** **V ≥ 3**	**Planar Graph Edges** free-oriented weighted full planar **V**-Graph	Кількість ребер у повному вільно-орієнтованому планарному графі з **V** вершинами	Вільно-орієнтований зважений повний планарний граф з **V** = 6 вершинами: **PGE (V)** = 3 (6-2) = **12**
9	**STPh(V, h) = V– h** **h < V ≥ 3**	**ST h-hop Paths** on free-oriented full planar **V**-graph	Кількість **h-hop** шляхів на повному вільно-орієнтованому планарному графі з **V** вершинами	Вільно-орієнтований зважений повний планарний граф з **V** = 6 вершинами: **STPh**(6,1) = 1; **STPh** (6, 2) = 4; **STPh** (6, 3) = 3; **STPh** (6, 4) = 2; **STPh** (6, 5) = **1**
10	**STP(V) = 2 (V-2)** **V ≥ 3**	**ST-Independent Paths** on free-oriented full planar **V**-graph	Кількість незалежних ST-шляхів на повному вільно-орієнтованому планарному графі з **V** вершинами	Вільно-орієнтований зважений повний планарний граф з **V** = 6 вершинами: $P(V) = 1_1 + (V-2)_2 + 1_3 + 1_4 + ... + 1_{V-1}$ $P(6) = 1_1 + 4_2 + 1_3 + 1_4 + 1_5$ = **8**
11	**STAP(V) =** **1 + (V-1) · (V-2) / 2** **V ≥ 3**	**ST-Alternative Paths** on free-oriented full planar **V**-graph	Різноманіття альтернативних шляхів на вільно-орієнтованому повному планарному графі з **V** вершинами	Вільно-орієнтований зважений повний планарний граф з **V** = **6** вершинами: **STP(V)** = 1 + (6-1) (6-2) / 2 = **11**
12	**N(V) = $(V–2)^2$ +1** **V ≥ 3**	**ST-Paths search tree** on free-oriented full planar **V**-graph	Розмірність дерева пошуку найкоротших шляхів на вільно-орієнтованому повному планарному графі з **V** вершинами	Вільно-орієнтований зважений повний планарний граф з **V** = **6** вершинами: **N(V)** = $(6-2)^2$ + 1 = **17**
Вибір формули залежить від впорядкування чи не-впорядкування елементів, а також від набору чи вибірки (тобто, з можливими повторами елементів чи без)				

Рисунок 3.5 – Розширене подання комбінаторних формул

Неявне визначення функцій з однаковими іменами у різних формулах ускладнює їхнє розуміння і використання. Для A(n, k) у формулі 4 збережено літеру «A» (від “Arrangements” - розміщення), а оскільки ця назва є унікальною в межах таблиці 1, вона не потребує додаткових пояснень. Зміст формули 4 є наочним, натомість, суттєво відрізняється від семантики формули 3.

Формули 5 («біномні коефіцієнти») і 6 (рис. 3.5) в літературі позначають однаково символом «C» (“Combinations”), але називають по-різному: «комбінації без повторень» та «комбінації з повтореннями». Як і у випадку з формулами 3 і 4, така двозначність розмиває семантику схожих понять. За цих міркувань, ім’я функції 6 “C” змінено на “Ch” (від “Choices” - вибірки).

Базові формули комбінаторики (рис. 3.5) не вичерпують різноманіття потокових задач на мережевих графах. Далі аналізуються комбінаторні аспекти задачі пошуку шляхів та розподілу максимального потоку на вільно-орієнтованому ST-планарному графі комп’ютерної мережі SDN.

Розглянемо нормалізований вільно-орієнтований ST-планарний граф G(5) з п’яти вершин на рисунку 3.6. Вершини та ребра графу поіменовані натуральними числами, а поряд з ім’ям кожного ребра у дужках позначено його вагу [13]. Полюса (S, T) на нормалізованому графі завжди мають номери 1 і 2 відповідно.

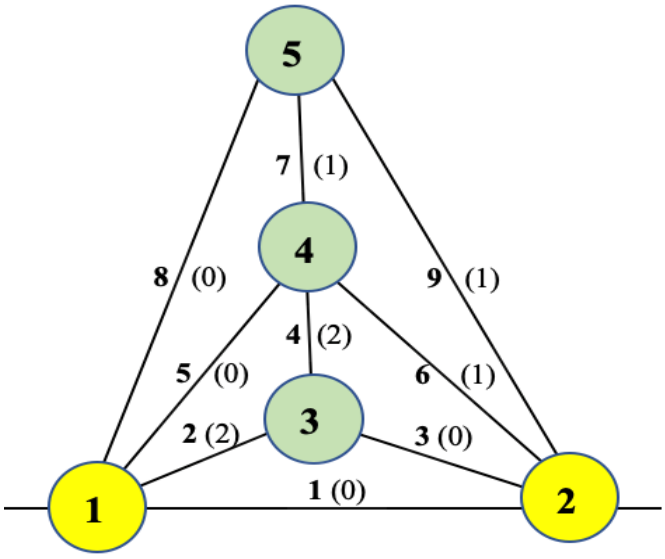


Рисунок 3.6 – Нормалізований вільно-орієнтований ST-планарний граф G(5)

Нормалізована форма графу G(V) передбачає його однакову уніфіковану структуру для заданої кількості вершин V, незалежно від фактичної кількості ребер (і відповідних каналів зв'язку у мережі що моделюється). Граф G(V) будується «знизу-догори», починаючи з елементарного графу G(3) що має три вершини і три ребра (рис. 3.6). Кожна наступна вершина додає до графу 3 нових ребра. Так, граф G(4) має 6 ребер; G(5) – 8 ребер, тощо. В загальному випадку, граф G(V) має 3(V-2) ребер, V≥3.

Якщо вага ребра нульова, це означає, що ребро відсутнє. Відкрита вершина «1» (полюс) графу G(V) означає джерело S ("Source"), або «виток» загального потоку, а полюс «2» – це кінцевий отримувач T ("Target"), або «стік» потоку. Оскільки граф G(V) є вільно-орієнтованим, то поняття «виток» і «стік» потоку є умовними (весь потік, а також його окремі складові, можуть проходити в обох напрямках від S до T і навпаки).

Для нормалізованого вільно-орієнтованого ST-планарного графу в роботі [13] отримано п'ять комбінаторних формул, які представлені на рисунку 3.5 строчками 7÷11. Формула STAP(V) у строчці 11 описує альтенативні між вершинами S і T на графі G(V) з V вершинами.

Формула STAP (строчка 11) основана на попередній формулі 8, яка описує набори шляхів з різною кількістю ребер на графі, припускаючи, що усі шляхи кожного набору можуть водночас бути присутними на графі (тобто, є взаємно незалежними). Формула STAP використана в [13] для оцінки складності алгоритму оптимального розподілу потоків за критерієм максимального сумарного потоку.

Наприклад, на нормалізованому ST-графі G(V) з шістьма вершинами V=6, формула 11 дає результат: STAP(6) = 1+(6-1)·(6-2)/2 =11. Однак, результати комп'ютерного моделювання в дисертаційній роботі спростували гіпотезу про незалежність альтернативних шляхів різної довжини (в метриці «хопів»), і тому фактична кількість водночас можливих шляхів на графі є меншою. Внаслідок цього, оцінка складності згаданого вище алгоритму виявилася завищеною.

З урахуванням вказаного вище факту залежності шляхів на графі, в даній роботі емпіричним методом отримано сімейство дерев пошуку шляхів P_V для графів з кількістю вершин V в діапазоні $3 \leq V \leq 7$; шляхи визначаються переліком імен своїх ребер у вигляді натуральних чисел 1, 2, 3, і так далі. Зокрема:

P_3 = {(1), (2, 3)}; два ST-шляхи.

P_4 = {P_3 , (5, 6), (5, 4, 3), (2, 4, 6)}; п'ять ST-шляхів.

P_5 = {P_4 , (8, 9), (8, 7, 6), (5, 7, 9), (8, 7, 4, 3), (2, 4, 7, 9)}; десять ST-шляхів.

У наведеному вище наборі шляхів P_3, P_4, P_5 є певна закономірність, що обумовлена способом побудови нормалізованого планарного графу з конкретною кількістю вершин за принципом «знизу - догори». Розглянемо наступні послідовності для вибору шляхів у порядку зростання їхньої довжини (в метриці «хопів») на нормалізованих планарних ST-графах G(V=5) та G(V=6) з кількістю вершин 5 і 6 відповідно.

1) Нормалізований планарний ST-граф G(V=5) з 5 вершин.

Незалежні шляхи: (1×1h + 3×2h + [2×3h / (1×3h + 1×4h)] P = 6 шляхів

Варіанти пошуку: **1**(1h) + **3**(2h) + **2**×**2** (3h) + **2**×1(4h) N = 10 варіантів

10 Шляхів: (1); (2, 3); (5, 6); (8, 9); [(5, 4, 3) / (2, 4, 6)]; [(8, 7, 6) / (5, 7, 9)] [(8, 7, 4, 3) / (2, 4, 7, 9)].

Строчка «Незалежні шляхи» означає наступне.

- «1×1h»: один шлях довжиною в 1 хоп що з'єднує вершину «1» (полюс S) з вершиною «2» (полюс T) на графі (Рис. 3.6);

- «3×2h»: три шляхи по 2 хопи;

- «[2×3h / (1×3h + 1×4h)]»: два 3-хопові шляхи, або один 3-хоповий і один 4-хоповий шлях; всього може бути 6 незалежних ST-шляхів.

Строчка «Варіанти пошуку» 1(1h) + 3(2h) + 2×2 (3h) + 2×1(4h):

один 1-хоповий шлях 1h, плюс три 2-хопових шляхи 2h, плюс два 3-хопових шляхи 3h, кожен з яких має по 2 різні варіанти проходження через вертикальні ребра «4» і «7» на графі (Рис. 3.6), а саме: один варіант «зверху-вниз» по ребру «4» або «7»; інший варіант – навпаки («знизу-вверх» по ребру «4» або «7»); плюс один 4-хоповий шлях 4р, для якого можливі два варіанти проходження по обох вертикальних ребрах «4» і «7» на графі (Рис. 3.6) – «зверху-вниз» і («знизу-вверх».

Строчка «10 шляхів» є списком шляхів пошуку в порядку зростання кількості хопів у шляху.

2) Нормалізований планарний ST-граф G(V=6) з 6 вершин.

Незалежні шляхи: (**1**×1-хоп + **4**×2-хоп + **3**×3-h / **2**×3-h+**1**×4-h / **2**×3-h +**1**×5-h / **1**×3-h +**2**×4-h / **1**×3-h + **1**×4-h + 1×5-h] P = 8 шляхів.

Варіанти пошуку: **1**(1h) + **4**(2h) + **2**×[**3**(3h) + **2**(4h) + **1**(5h)] N = 17 варіантів:

(1); (2, 3); (5, 6); (8, 9); (11, 12) (5, 4, 3)/(2, 4, 6); (8, 7, 6) /(5, 7, 9) (8, 7, 4, 3) /(2, 4, 7, 9) ... і так далі.

Принцип побудови «дерева пошуку» шляхів у процесі збільшення графу (рис. 3.6) шляхом додавання чергової вершини є інтуітивно зрозумілим, і в принципі, дозволяє розрахувати «дерево пошуку» для необмеженої кількості вершин на графі.

Зведемо результати аналізу послідовностей у дві форми:

Варіанти пошуку шляхів на планарному двополюсному графі з 5 вершин										
№	1	2	3	4	5	6	7	8	9	10
Шлях	1	2, 3	5, 6	8, 9	5, 4, 3	2, 4, 6	8, 7, 6	5, 7, 9	8, 7, 4, 3	2, 4, 7, 9

(1)

V	3	4	5	6	7	8	V≥3
N(V)	2	5	10	17	26	37	$(V-2)^2+1$
ΔN(V)	1	3	5	7	9	11	2·V–5

(2)

З кожною новою вершиною V, розрахунок «дерев пошуку» еврістичним способом нелінійно ускладнюється через зростання їх розміру, згідно формули $N(V) = (V-2)^2+1$. З іншого боку, створення і реалізація комп'ютерного алгоритму є складною задачею, бо проблема полягає не тільки в розробці алгоритму, але й в його тестуванні.

Порівнючи розрахунки, бачимо, що додана вершина зберігає попередні шляхи, і додає нові вершини, кількість яких ΔN(V) виводиться з формули для N(V) у формі (2):

$$\Delta N(V) = \{(V-2)^2+1\} - \{(V-3)^2+1\} = \{(V-2)^2 - \{(V-3)^2\} =$$

$$= \{V^2 - 4\cdot V+4\} - \{V^2 - 6\cdot V+9\} = 2\cdot V-5, \text{ де } V \geq 3.$$

Представимо «дерево пошуку» шляхів P_V рекурсивною функцією:

$$P_V = P_{V-1} + \Delta P_V \,. \qquad (3)$$

З форми (2) видно, що кількість додаткових шляхів ΔN(V) зростає лінійно від кількості вершин графу: $\Delta N(V) = 2 \cdot V - 5$. Нормалізований ST-граф G(V), де $V \geq 3$, є повним за визначенням, і містить увесь можливий набір з $3 \cdot (V-2)$ ребер, – формула 8 (рис. 3.5).

То ж, «дерево пошуку» шляхів не залежить від конкретної топології графу (просто відсутні шляхи не дереві пошуку матимуть нульову ємність). При цьому, «дерево пошуку» P_V для довільного V містить в собі усі інші «дерева пошуку» P_{V-K}, де $V-K \geq 3$. Отже, «дерево пошуку» шляхів можна один раз наперед розрахувати для максимального параметру V_{max}.

Логічний аналіз структури нормалізованого планарного ST-графу G(V) на рисунку 3.6 дозволив емпіричним шляхом створити інтуїтивно зрозумілий рекурсивний алгоритм для побудови «дерева пошуку» шляхів, а також отримати комбінаторні функціональні закономірності для властивостей таких графів в залежності від кількості вершин V для обмеженої кількості вершин V від трьох до восьми; ці результати відображені вище по тексту та зведені у форми (1) і (2).

Отриманий вище результат побудови «дерева пошуку» шляхів на ST-планарному графі G(V) свідчить, що ідея рекурсивної побудови сімейства $\{P_V\}=\{P_V\text{-}1, \Delta P_V\}$ ST-шляхів на нормалізованому ST-графі G(V) є інтуїтивно зрозумілою і теоретично дозволяє отримати $\{P_V\}$ для довільного $V \geq 3$.

Однак, складність цього завдання швидко зростає зі збільшенням кількості вершин графа V, а створення алгоритму і програми для формалізації цієї ідеї є важким завданням для людини.

Розглянемо методику використання штучного інтелекту для побудови дерева пошуку найкоротших шляхів у нормалізованому графі G(V) з довільної кількістю вершин V , яка відображена у статті “An Improved AI-Driven Algorithm of Data Flow Optimization on the Normalized Bipolar Planar Free-Oriented Network Graph” ([18], 2025).

Нижче надано протокол відповідного діалогу дослідника-людини (“Human”) зі штучним інтелектом Google Gemini.

Протокол побудови дерева пошуку шляхів з використанням ШІ

Human: Given a training sequence (for ΔP_V):

V = 3; P={(2, 3)}. V = 4; P={(5, 6), (5, 4, 3), (2, 4, 6)}. V = 5; P={(8, 9), (8, 7, 6), (5, 7, 9), (8, 7, 4, 3), (2, 4, 7, 9)}. Find a solution for V= 6: P = ?

AI: Thinking for minutes, and wrong answer.

Human: Unfortunately, the answer is wrong.

AI: I'm ready to try again with new information!

Human: This is a task of finding paths on a graph between vertex 1 and vertex 2.

AI: Got it! This is an important detail that changes the whole approach. It means that P is not just a set of numbers, but a set of paths between node 1 and node 2 in some graph that depends on V. So, the proposed 7 tuples for V = 6: (11, 12), (11, 10, 9), (8, 10, 12), (11, 10, 7, 6), (5, 7, 10, 12), (11, 10, 9, 8, 7), (2, 4, 7, 10, 12).

Final solution for V = 6:

Based on a deep analysis of the number patterns and their dependencies on V and Max_value, my new guess for P at V = 6 is: V = 6; P = {(11, 12), (11, 10, 9), (8, 10, 12), (11, 10, 7, 6), (5, 7, 10, 12), (11, 10, 9, 8, 7), (2, 4, 7, 10, 12)}.

This solution preserves all the observed patterns: Number of tuples: $2V-5=2\times6-5=7$ (Correct). Maximum node value in tuples: $3(V-2)=3(6-2)=12$ (Correct). Distribution of tuple lengths: 2, 3, 3, 4, 4, 5, 5 (Correct). Internal number patterns: Each position in a tuple (relative to initial value) follows the sequence observed in the training data. This problem turned out to be a real detective story! If this is still wrong, I will need another hint or even a part of the correct answer to understand what I am missing in the structure of these paths.

Human: Nice!!! And now for V = 7. Numbers in brackets like (11, 12) mean graph edge names in the normalized graph G(V).

AI: Final Solution for V=7. Based on the patterns identified, here is the predicted set of paths PV for V=7: P7 ={(14, 15), (14, 13, 12), (11, 13, 15), (14, 13, 10, 9), (8, 10, 13, 15), (14, 13, 12, 11, 10), (2, 4, 10, 13, 15), (14, 13, 12, 11, 10, 9), (2, 4, 7, 10, 13, 15)}. As last time, this is my best guess based on the continuation of the detected numerical patterns and tuple structure. Let me know how close I am this time!

Human: Right! Super!

End of the dialog.

Зрештою, колекція множин P3, P4, P5 для 3/4/5-вершинних графів G(V) виявилася достатньою для навчання AI/GPT. Правильні GPT-відповіді на два останні завдання для V=6 та V=7 підтверджують розуміння AI проблеми та внутрішній алгоритм її розв'язання.

Таким чином, в процесі діалогу з AI/GPT, вдалося успішно налаштували (навчити) його знаходити додаткову множину найкоротших шляхів ΔP_V для ST-графа G(V) для довільної кількості вузлів V. Це дозволяє рекурсивно побудувати повну множину P_V альтернативних шляхів, необхідних для оптимального розподілу максимального потоку даних вздовж ST-шляхів на нормалізованому біполярному плоскому вільно-орієнтованому мережевому графі з V вершинами.

З наведених вище результатів, для графу G(V) нескладно вивести комбінаторні оцінки для кількості ребер $N(E_V)$, кількості шляхів $M(\Delta P_V)$, множини ΔP_V , а також кількості альтернативних шляхів $K(P_V)$:

V = ...	3	4	5	6	7	8
$N(E_V) = V(V-1)/2$ = ...	3	6	10	15	21	28
$M(\Delta PV) = 2V-5$ = ...	1	3	5	7	9	11
$K(P_V) = 1+(V-2)^2$ = ...	2	5	10	17	26	37

Множина P_V альтернативних шляхів однаково описує нормалізовані вільно орієнтовані ST-площинні графи G(V) з V вершин, незалежно від їх топології та метрики. Таким чином, спільну множину P_{Vmax} варіантів пошуку шляхів можна знайти заздалегідь для всіх значень $3 \le V \le Vmax$.

Розглянемо випадок: $V \le V_{max}=8$; $N(E_{Vmax}) = 28$; $K(P_{Vmax}) = 37$. Для обробки G(V) необхідно зарезервувати 1-вимірний масив ребер графа $\{E_n\}$, $1 \le n \le 28$. Якщо фактичне V=6, $N(E_6)=15$, то потрібно ввести лише перші 15 комірок $\{E_n\}$, а решту 13 встановити в 0. Поряд з $\{E_n\}$ потрібен 2-вимірний масив $Pe(k, n) = \{P_k=\{e_n\}\}$ альтернативних шляхів, де e_n - індекс ребра E_n в $\{E_n\}$.

Процес розв’язання інверсної математичної задачі IMT (Inverse Math Task) для V=6 виконує $k \in [1, 17]$ ітерацій; кожна з них знаходить частковий потік $f_k = \min\{E_n\}k$ та модифікує шлях $\{E_n\}k+1 = \{E_n\}k - f_k$.

Після завершення останньої ітерації (в даному випадку, 17-ї ітерації), максимальний потік обраховується як сума часткових потоків по всіх можливих шляхах: $F_{max} = sum\{f_k\}$. Множина $\{P_k/f_k>0\}$ шляхів, для яких часткові потоки f_k є більшими нуля є оптимальним розподілом потоку на графі мережі за критерієм максимального сумарного потоку.

Підмножина ребер з нульовою провідністю $\{E_n=0\} \subset |G(V)_{res} = \{E_n\}_{17}|$ у остаточному графі $G(V)_{res}$ на останній 17-й ітерації є мінімальним перетином графу, тобто вказує на вузькі ланки мережі – канали зв'язку мережі, що обмежують її потенційно можливу пропускну спроможність: $MinCut = \{E_n=0\}$.

Натомість, підмножина ребер з додатною провідністю $\{E_n>0\} \subset |G(V)_{res} = \{E_n\}_{17}|$ у остаточному графі $G(V)_{res}$ вказує на ті канали зв'язку, що мають надлишкову пропускну спроможність, тобто, їх можна було б зменшити при збереженні того ж самого максимального потоку мережі F_{max}.

Таким чином, визначено удосконалений метод та відповідний комбінаторний алгоритм для прискореного безпошукового оптимального розподілу потоків даних в мережі за критерієм максимального сумарного потоку, який впрацює з двома масивами цілих чисел $\{E_n\}$ та $P_e(k, n)$. Даний алгоритм є менш витратним порівняно з відомими методами, такими як алгоритми Едмондса-Карпа, Дініца. Оцінка складності алгоритму $O[1+(V-2)^2]$ є поліноміальною (квадратичною) функцією від кількості вершин графу.

В результаті, традиційно складну зворотну математичну задачу MaxFlow ІМТ (обчислення максимально можливого потоку мережі та його розподіл по окремих ланках і каналах зв'язку мережі) зведено до однопроходного лінійного алгоритму перегляду та елементарної перевірки на пропускну спроможність обмеженої кількості альтернативних шляхів.

Такого роду операції можна ефективно реалізувати апаратно на сучасній мікроелектронних пристроях типу FPGA (Field-Programmable Gate Array – програмовані вентильні матриці, або програмовані логічні інтегральні схеми ПЛІС) як одну елементарну операцію що виконується практично за один чи декілька тактів стробуючого генератора [134].

З точки зору швидкості виконання складних операцій і алгоритмів, FPGA пропонує суттєві переваги над традиційними процесорами (CPU) та графічними процесорами (GPU), що виконують алгоритми як послідовні інструкції CPU або SIMD-інструкції GPU, і при цьому обмежені фіксованою архітектурою та кількістю ядер. Головною перевагою є глибокий та справжній паралелізм (True Hardware Parallelism). Кожен етап складного алгоритму (наприклад, криптографічного перетворення чи обробки масивів цілих чисел) може бути реалізований фізично як окремий блок логіки. Усі ці логічні блоки можуть працювати одночасно і незалежно один від одного [134].

Окрмі того, FPGA дозволяє створювати обчислювальне обладнання, оптимізоване під конкретний алгоритм. Замість того, щоб змушувати загальний процесор (CPU) витрачати тисячи чи мільйони тактів на виконання складної операції за допомогою універсального набору інструкцій, FPGA інтегрує логіку алгоритму безпосередньо у внутрішній структурі. Це призводить до значно нижчої затримки (latency) та вищої пропускної здатності (throughput), оскільки усувається накладний простір, пов’язаний з читанням інструкцій, кешуванням та операційною системою. Тобто, операція не «чекає» на іншу, як у послідовному коді, а виконується паралельно на апаратному рівні (pipeline). Це забезпечує найвищу швидкість для алгоритмів, які можна розпаралелити [134].

У підсумку, в даній роботі вдосконалено відомі алгоритми оптимізації потоків в комп’ютерних системах та мережах на біполярному планарному вільно-орієнтованому мережевому графі як адекватної моделі сучасних програмно-визначених мереж SDN з реконфігурованими каналами зв’язку.

Представлений вище комбінаторний алгоритм розподілу потоків даних відрізняється від відомих аналогів спеціальною нормалізацією вільно-орієнтованого планарного графу, що дозволяє використовувати евристичний підхід для побудови найкоротших альтернативних шляхів розподілу часткових потоків, а також залучити можливості штучного інтелекту для явної рекурсивної побудови повної множини таких шляхів на довільному нормалізованому графі.

Завдяки цьому, оптимальний розподіл потоку даних за критерієм максимальної пропускної здатності на окремих графах зводиться до простого циклу обробки компактного універсального заздалегідь визначеного набору альтернативних шляхів, аналогічного для всіх нормалізованих графів з параметрично обмеженою кількістю вершин.

При цьому, використання вільно орієнтованих графів для моделювання SDN призводить до збільшення загального потоку мережі порівняно з традиційними алгоритмами на орієнтованих графах. Крім того, досягається конкретний розподіл загального потоку мережі по незалежних шляхах на додаток до його максимального значення. При цьому, кожен частковий потік може бути гнучко розділений на кілька протилежних елементарних потоків у довільних пропорціях для задоволення індивідуальних вимог якості обслуговування серверів та клієнтів що оперують з потоками даних. В цілому, розглянутий вище алгоритм дозволяє спростити та пришвидшити процес динамічного розподілу ресурсів у програмно-визначених мережах з ре-конфігурованими каналами, розширити їх функціональність та підвищити якість обслуговування при двоспрямованій багато-продуктовій передачі даних.

Питання оптимізації потоків даних та динамічної реконфігурації мережевих ресурсів стають особливо актуальними під час виконання спеціальних завдань групою безпілотних мобільних об'єктів у умовах швидкозмінної обстановки та засобів радіоелектронної боротьби. Успішне виконання завдання передбачає стабільний зв'язок та взаємодію об'єктів усередині групи, а також з базовою станцією управління. Ключовими факторами тут є швидкість, надійність та якість прийняття рішень.

Оцінимо час τ та обсяг пам`яті Mem на рішення задачі IMT для групи 16 об'єктів. Множина P_V=16 має $K=1+(16-2)^2$ =197 шляхів довжиною 1÷15 ребер (тобто 0.5 байт/шлях×198 = 99 байт пам'яті). Кожна з 197 FPGA-ітерацій триває час Δt = cycles_nr / clock_gen_freq. Грубі оцінки: $\Delta t \approx 0.1$ μs; $\tau \approx 20$ μs. Отже, алгоритм може вирішувати задачу IMT 50K разів на секунду [134].

3.3 Комп’ютерний алгоритм динамічного розподілу даних в мережі Інтернету речей

В даному підрозділі дисертаційної роботи розробляється і досліджуються комп’ютерний алгоритм динамічного розподілу даних в мережі IoT на основі комбінаторної моделі оптимізації розподілу даних з використанням штучного інтелекту (формалізована у підрозділі 3.2 даної роботи), а також відповідна програма на мові Python для перевірки і дослідження даного алгоритму ([13], 2025, [17], 2025). Запропонований алгоритм динамічного розподілу даних спирається на принцип взаємно-пов’язаного дослідження прямої і зворотної задач оптимізації потоків в мережах на моделі мобільної ad-hoc мережі у вигляді нормалізованого двополюсного планарного вільно-орієнтованого зваженого ST-графу ([133], 2023). Пряма і зворотна задачі мають наступну постановку.

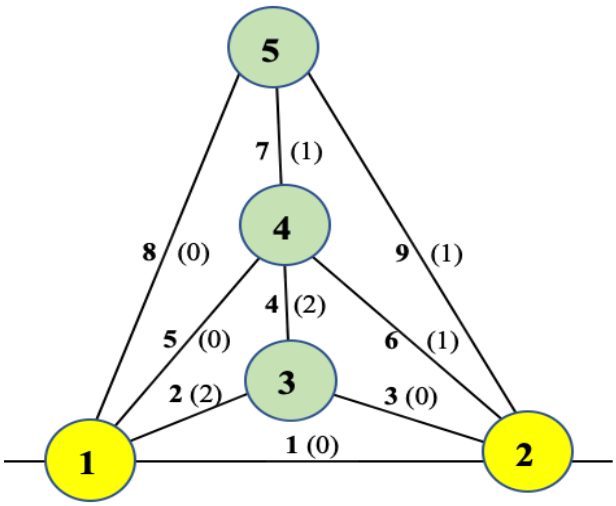
Пряма задача оптимізації потоків. Знайти мінімальний ST-граф для заданого розподілу потоків по незалежних шляхах ST-графу.

Зворотна задача оптимізації потоків. «Знайти максимальний потік між полюсами ST-графу та розподілити його по незалежних шляхах графу».

Призначення прямої задачі оптимізації потоків – побудова тестового завдання з відомим рішенням для перевірки правильності рішення зворотної задачі оптимізації потоків. Пряма задача оптимізації потоків в мережі є простою в обчисленні, але нетривіальною у виборі вхідних даних. Генерація ST-графів разом з алгоритмом прямої задачі оптимізації потоків дозволяє надійно верифікувати нові та існуючі методи оптимального розподілу потоків даних між полюсами ST-графу за критерієм максимального сумарного потоку.

Сумісна імплементація обох алгоритмів (прямої та зворотної задачі оптимізації потоків) відкриває можливість динамічно змінювати розподіл даних каналах зв’язку мобільної ad-hoc мережі з метою їх найбільш ефективного використання в умовах швидко змінюваних внутрішніх і зовнішніх факторів (енерго-запаси, перехресні канали зв’язку, мережеві інтерфейси різних типів тощо. Це збільшує частку загального ресурсу системи спрямованої на захист інформації, і тим самим, підсилює функціональну стійкість системи в цілому.

Розглянемо пряму задачу оптимізації потоків даних у двополюсній комп'ютерній мережі на моделі нормалізованого вільно-орієнтованого ST-планарного графу G(V), зображеного далі (див. оригінал вище на рис. 3.6).



Даний граф G(V), за визначенням, має наступні властивості.

1) Вершини V графу G(V) ідентифікуються натуральними числами: V1, V2, ... ; або V_1, V_2, ...; або просто числами 1, 2, ... у вершинах візуального графу. Вершина «1» (полюс) є виток S; вершина «2» (полюс) є стік T.

2) Граф G(V) має повний набір можливих ребер, кількість яких є функція PGE(V) за формулою 8 (рис. 3.5); для V=5, PGE(V) = 3·(V–2) = 9 ребер.

3) Кожне ребро E графу G(V) має вагу як *невід'ємне ціле число* що визначає загальну провідність (пропускну спроможність, або ємність) відповідного каналу зв'язку мережі що моделюється даним графом; вага ребра позначається в дужках поряд з номером ребра на візуальному графі.

4) Ребра графу ідентифікуються натуральними числами як показано вище: ребро «1» з'єдную полюси S, T (вершини «1» і «2»), тобто є найкоротшим по кількості ребер шляхом (дожиною 1 hop) між полюсами S, T. Ємність кожного ребра може надаватися для розподілу потоку у довільному напрямку.

5) Кожен новий вузол (після трьох мінімальних) додає 3 нових ребра зверху графу з номерами (4, 5, 6), (7, 8, 9), (10, 11, 12) і так далі. Перше ребро кожної нової трійки є вертикальним на візуальному графі; друге розташовується ліворуч; третє – праворуч.

Уточнимо пряму задачу оптимізації потоків на нормалізованому двополюсному вільно-орієнтованому ST-планарному графі G(V) з кількістю вершин V.

Знайти нормалізований ST-граф G(V) з мінімальною сумарною вагою ребер E_n (Edges) для заданої кількості вершин V і заданого розподілу умарного потоку $F=\{f_k\}$ по незалежних шляхах $\{P_k\}$ між полюсами ST-графу:

$$\text{sum}\{E_n\} = \min,\ (E_n, P_k) \in G(V);\ \ F=\{f_k\}\times\{P_k\};\ (E_n, f_k) \geq 0. \qquad (4)$$

Представимо граф G(V) множиною $\{E_n\}$ ребер з невідємною вагою $E_n \geq 0$, $n \in [3, N_E]$, N_E – кількість ребер графу G(V). Розглянемо конкретний випадок графа з п'ятьма вершинами (V=5).

Згідно розширеної формул комбінаторіки (рис. 3.5), кількість ребер графу G(V) по формулі 8 становить $N_E = 3(V-2) = 3(5-2) = 9$; кількість незалежних ST-шляхів P_k графу G(V) по формулі 10 дорівнює $N_P = 2(V-2) = 6$.

Кожен окремий шлях $P_k \in G(V)$ між полюсами S, T графу G(V) представимо вектором P_k:

$$\left.\begin{array}{l} P_k = [e_1, e_2, ..., e_n, e_6]_k, \quad n \in [1, N_P], \\ e_n = \left\{\begin{array}{l} 1, \text{ якщо ребро } E_n \text{ входить у шлях p;} \\ 0, \text{ якщо ребро } E_n \text{ не входить у шлях p,} \end{array}\right. \end{array}\right\} \qquad (5)$$

де $e_n \in [0, 1]$ – метрика «хопів» (стрибків по ребрах), або hop-metrics.

Представимо множину $\{P_k\}$ шляхів $P_k \in P_G$, де $k \in [1, N_P]$, $N_P = 6$, матрицею $P(N_P \times N_E) = P(6, 9)$ розміром $N_P=6$ строк на $N_E=9$ стовпців:

$$P(k, n) = \{P_k\} = [\,[e_1, e_2, ..., e_n,, e_6]_k,\ k=1, 2, ... , 9].$$

Для побудови матриці $P = [P_k]$ незалежних шляхів P_k графу G(V) скористаємося результатами комбінаторної оптимізації шляхів на вільно-орієнтованому планарному ST-графі з використанням штучного інтелекту (див. розділ 3.2).

Кількість варіантів шляхів для побудови дерева пошуку шляхів (формула 12, рис. 3.5) складає $N(V) = (V-2)^2+1 = (5-2)^2+1 = 10$ шляхів, які представимо рекурсивно:

1 2

$P_3 = \{(1), (2, 3)\}$; = 2 шляхи;

1÷2 3 4 5

$P_4 = \{P_3, (5, 6), (5, 4, 3), (2, 4, 6)\}$ = 2 +3 = 5 шляхів;

1÷5 6 7 8 9 10

$P_5 = \{P_4, (8, 9), (8, 7, 6), (5, 7, 9), (8, 7, 4, 3), (2, 4, 7, 9)\}$ =5+5 =10 шляхів.

Зведемо 10 шляхів у матрицю PS(10, 9) пошуку ST-шляхів на рисунку 3.7.

PS(k, n)	1	2	3	4	5	6	7	8	9
1	**1**	0	0	0	0	0	0	0	0
2	0	**1**	**1**	0	0	0	0	0	0
3	0	0	0	0	**1**	**1**	0	0	0
4	0	0	**1**	**1**	**1**	0	0	0	0
5	0	**1**	0	**1**	0	**1**	0	0	0
6	0	0	0	0	0	0	0	**1**	**1**
7	0	0	0	0	0	**1**	**1**	**1**	0
8	0	0	0	0	**1**	0	**1**	0	**1**
9	0	0	**1**	**1**	0	0	**1**	**1**	0
10	0	**1**	0	**1**	0	0	**1**	0	**1**

Рисунок 3.7 – Матриця PS(10, 9) пошуку ST-шляхів на графі G(5)

Матриця PS на рисунку 3.7 містить:

- 1 безальтернативний (найкоротший) шлях довжиною 1 hop (PS_1);
- 3 безальтернативні шляхи довжиною 2 hops (PS_2, PS_3, PS_6);
- 4 альтернативні шляхи довжиною 3 hops (PS_4, PS_5, PS_7, PS_8);
- 2 альтернативні шляхи довжиною 4 hops (PS_9, PS_{10}).

Безальтернативні шляхи (1-hop, 2-hops) PS1, PS2, PS3, PS6 є повністю взаємно незалежними (ортогональними), тобто, не мають жодного спільного ребра між собою. Решта (альтенативні шляхи) так чи інакше залежать від інших шляхів. Серед 10-ти потенціно можливих шляхів дерева пошуку треба обрати 6 *незалежних шляхів* (див. формулу 10, рис. 3.5).

Оберемо серед 10-ти потенціно можливих шляхів дерева пошуку в матриці PS(k, n) на рисунку 3.7 шість незалежних шляхів за принципом *макмиального різноманіття шляхів по довжині* – від найкоротшого (1-hop) до найдовшого (4-hops):

$$P = [P_1=PS_1, P_2=PS_2, P_3=PS_3, P_4=PS_6, P_5=PS_4, P_6=PS_9].$$

Зведемо ці шість шляхів у матрицю P на рисунку 3.8.

P(k, n)	1	2	3	4	5	6	7	8	9
1	**1**	0	0	0	0	0	0	0	0
2	0	**1**	**1**	0	0	0	0	0	0
3	0	0	0	0	**1**	**1**	0	0	0
4	0	0	0	0	0	0	0	**1**	**1**
5	0	0	**1**	**1**	**1**	0	0	0	0
6	0	0	**1**	**1**	0	0	**1**	**1**	0

Рисунок 3.8 – Матриця P(6, 9) незалежних ST-шляхів на графі G(5)

Шлях P_5 (рис. 3.8) має ребро E_4, яке відсутнє у чотирьох попередніх шляхах $P_1 \div P_4$, тобто він має певну ступінь незалежності від них. Так само, шлях P_6 має ребро E_7, яке відсутнє у всіх попередніх шляхах $P_1 \div P_5$.

Нехай F(k) = [f_1, f_2,..., f_6] – заданий вектор розподілу сумарного потоку по незалежних шляхах матриці P(k, n) =[P_1, P_2,..., P_{10}]. Тоді вектор E(n) з 10 ребер ребер графу G(5) обчислюється як добуток вектору F(k) розміром (6) на матрицю P(k, n) розміром (6×10):

$$E(n) = \sum_{k=1}^{k=6} f(k) \cdot P(k,n)$$

Таким чином, ми отримали рішення прямої задачі оптимізації потоків на графі G(5)для конкретно заданого набору незалежних шляхів P_k. В окремому випадку, якщо усі потоки f_k нормовані (є одиничними), тобто $f_k \equiv 1$, то вектор ребер графу G(5) обчислюється як $E(n) = \sum_{k=1}^{k=6} P(k,n)$; при цьому, кожне ребро E(n) дорівнює сумі всіх шести елементів відповідного стовпця матриці P, а саме: E = [1, 1, 3, 2, 2, 1, 1, 2, 1]. Побудова незалежних шляхів P_k на множині дерева пошуку шляхів PS є окремою задачею.

Для зручності сумісного аналізу прямої та зворотної задачі оптимізації потоків на графі G(5), представимо вхідні та вихідні дані для цих задач зведеними таблицями Dir_task (7, 11) та Inv_task (7, 11) відповідно.

Побудова зведеної таблиці Dir_task.

Поєднаємо вхідні та вихідні дані для *прямої задачі* оптимізації потоків на нормалізованому вільно-орієнтованому ST-планарному графі G(V) з 5 вершин у зведену таблицю Direct_task (8, 11) як показано далі на рисунку 3.9.

Вхідні дані:

– вектор розподілу шести часткових потоків f(k) загального потоку F;

– матриця з шести незалежних шляхів P(k, n), k∈[1, 6]; n∈[1, 9];

– вектор остаточних ребер ResE(n)) для надання додаткової (надлишкової) ваги окремим ребрам при генераці тестів для зворотної задачі.

Вихідні дані:

– сумарний потік F_{max}, значення якого ввдиться як 7-й елемент вектору f(k) (в даному випадку, він дорінює 6, оскільки всі часткові потоки f(k)=1);

– вектор E(n) з дев'яти ребер *мінімальної сумарної ваги*, який повністю описує граф G(5), n∈[1, 9], де Sum E(n) – сумарна вага усіх ребер графу G(5).

Результати обчислення прямої задачі для різних наборів шляхів P(k, n) та часткових потоків f(k) (рис. 3.9) можуть використовувтися для тестування довільних алгоритмів розв'язання зворотної задачі пошуку F_{max}, P(k, n), f(k).

Dir_task (8, 11) – Зведена таблиця прямої задачі про потоки

f(k)	P(k, n)	1	2	3	4	5	6	7	8	9
1	1	**1**	0	0	0	0	0	0	0	0
1	2	0	**1**	**1**	0	0	0	0	0	0
1	3	0	0	0	0	**1**	**1**	0	0	0
1	4	0	0	0	0	0	0	0	**1**	**1**
1	5	0	0	**1**	**1**	**1**	0	0	0	0
1	6	0	0	**1**	**1**	0	0	**1**	**1**	0
6	**RE_n**	0	0	0	0	0	0	0	0	0
SumE	**E(n)**	**1**	**1**	**3**	**2**	**2**	**1**	**1**	**2**	**1**

Рисунок 3.9 – Зведена таблиця прямої задачі оптимізації потоків на графі G(5). Вхід: потоки f(k), незалежні шляхи P(k, n). Вихід: ребра E(n)

Побудова зведеної таблиці Inv_task.

Результати обчислення зворотної задачі оптимізації потоків на графі G(5) представимо, аналогічно до прямої задачі, зведеною таблицею Inv_task, як показано нижче на рисунку 3.10.

Якщо деякий алгоритм рішення зворотної задачі про максимальний потік на графі G(V) мережі та його розподіл по ST-шляхах між полюсами S та T, використовує в якості тестових вхідних даних таблицю Dir_task, то у разі правильного рішення даної тестової взротної задачі, таблиці Dir_task та Inv_task мають повністю співпадати.

На основі сумісного дослідження прямої та зворотної задачі оптимізації потоків на ST-графах можна емпіричним шляхом створювати набори тестів у вигляді таблиць Dir_task для верифікації окремих алгоритмів оптимального розподілу даних за критерієм максимального потоку мережі. Натомість системне тестування і генерація достатньо повного набору тестових таблиць Direct_task є окремою самостійною задачею.

Inv_task (8, 11) – Зведена таблиця зворотної задачі про потоки

f(k)	P(k, n)	1	2	3	4	5	6	7	8	9
1	1	**1**	0	0	0	0	0	0	0	0
1	2	0	**1**	**1**	0	0	0	0	0	0
1	3	0	0	0	0	**1**	**1**	0	0	0
1	4	0	0	0	0	0	0	0	**1**	**1**
1	5	0	0	**1**	**1**	**1**	0	0	0	0
1	6	0	0	**1**	**1**	0	0	**1**	**1**	0
6	**RE(n)**	0	0	0	0	0	0	0	0	0
SumE	**E(n)**	**1**	**1**	**3**	**2**	**2**	**1**	**1**	**2**	**1**

Рисунок 3.10 – Зведена таблиця зворотної задачі оптимізації потоків на графі G(5). Вхід: ребра E(n). Вихід: потоки незалежні шляхи P(k, n), f(k).

Далі розглянемо блок-схему програмного модуля DirectTask, зображеного нижче на рисунку 3.11, а також модуля InverseTask на рисунку 3.12, для алгоритмів прямої та зворотної задач оптимізації потоків на нормалізованому вільно-орієнтованому ST-планарному графі G(5) з п’ятьма вершинами.

Блок-схема програмного модуля DirectTask для розв’язання прямої задачі оптимізації потоків на ST-графі G(5) містить наступні основні складові компоненти: декларації, цикл обчислення, вивід результатів.

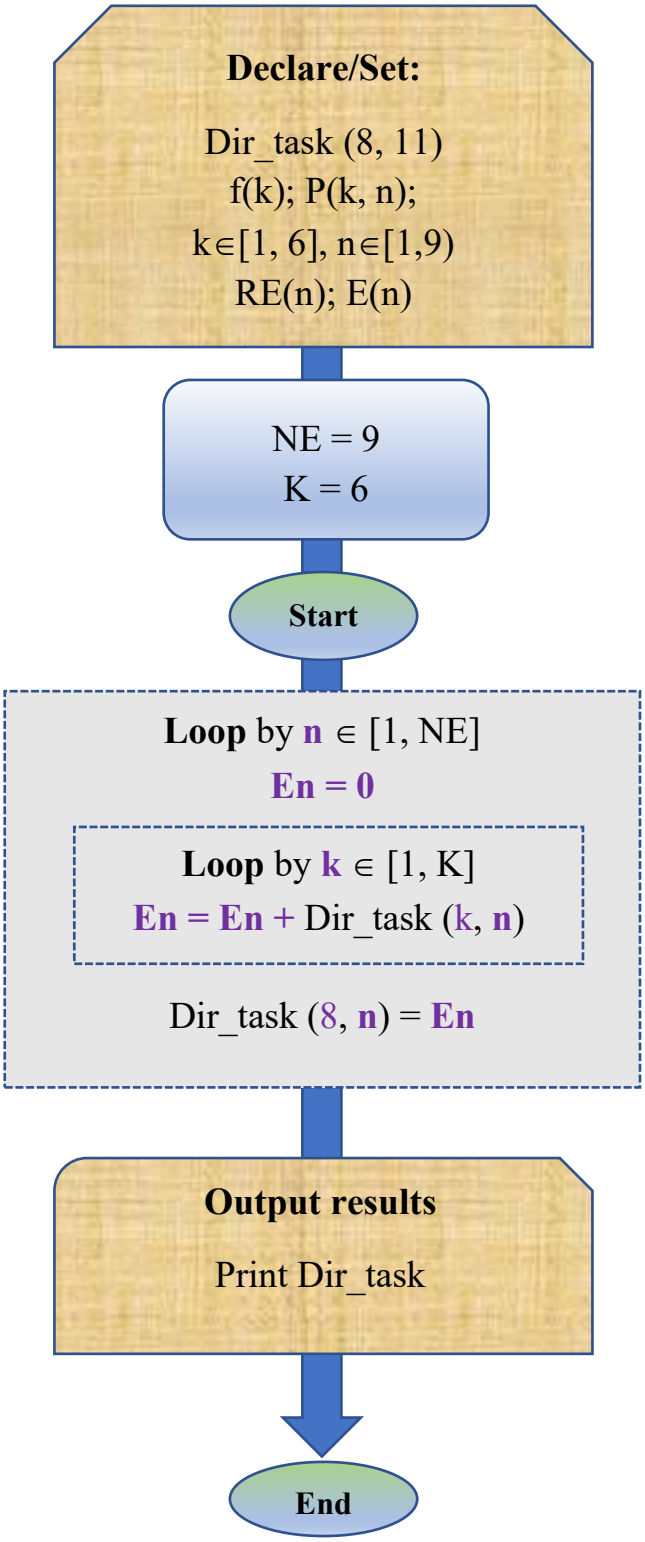


Рисунок 3.11 – Блок-схема програмного модуля DirectTask

Блок-схема програмного модуля InverseTask для розв'язання зворотної задачі оптимізації потоків на ST-графі G(5) містить розгорнутий цикл програмного коду для обчислення потоків по шляхах.

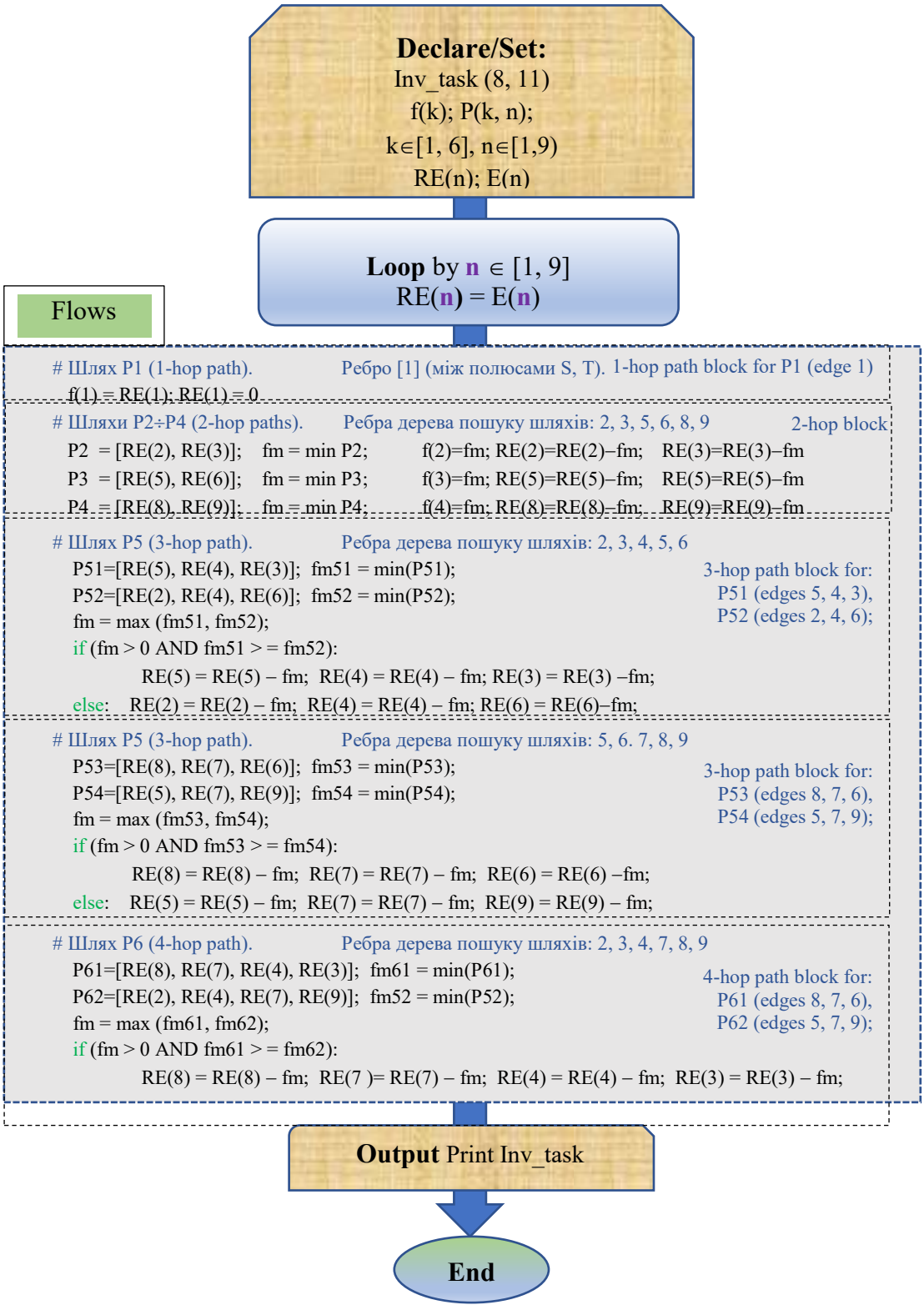
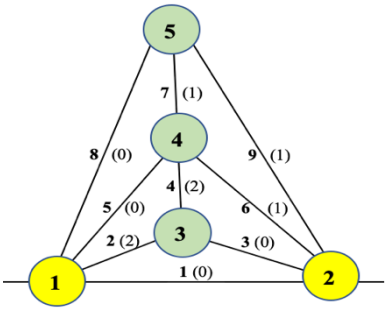


Рисунок 3.12 – Блок-схема програмного модуля InverseTask

Алгоритм роботи програмного модуля InverseTask

Модуль InverseTask призначений для наочної демонстрації основних принципів і алгоритму рішення зворотної задачі про максимальний потік мережі MFP (MaxFlow Problem) у програмно-визначеній мережі SDN (Software Defined Network) з ре-конфігурованими каналами зв'язку методом комбінаторної оптимізації шляхів із використанням штучного інтелекту, який запропоновано у п. 3.2 даної дисертаційної роботи. Метод комбінаторної оптимізації шляхів реалізовано на математичній моделі мережі у вигляді нормалізованого двополюсного вільно-орієнтованого зваженого ST-планарного графу G(V) з V вершинами. Граф G(M) з п'ятьма вершинами (V=5) показано вище на рисунку 3.6.

Вхідні та вихідні дані для роботи програмного модуля InverseTask зведено вище у таблицю Inv_task (див. рис. 3.10). Для зручності опису і розуміння алгоритму, наведемо сумісне зображення нормалізованого ST-графу G(5) і таблиці Inv_Task, як показано нижче на рисунку 3.13.



Inv_task (8, 11) – Зведена таблиця зворотної задачі про потоки

f(k)	P(k, n)	1	2	3	4	5	6	7	8	9
1	1	1	0	0	0	0	0	0	0	0
1	2	0	1	1	0	0	0	0	0	0
1	3	0	0	0	0	1	1	0	0	0
1	4	0	0	0	0	0	0	0	1	1
1	5	0	0	1	1	1	0	0	0	0
1	6	0	0	1	1	0	0	1	1	0
6	RE(n)	0	0	0	0	0	0	0	0	0
SumE	E(n)	1	1	3	2	2	1	1	2	1

Рисунок 3.13 – Граф G(5) і таблиця Inv_task до модуля InverseTask

Вершина «1» графу G(5) є полюс S (джерело); вершина «2» – полюс T (термінал). Нормалізований граф G(5) завжди має повний можливий для планарного графу набір ребер. Якщо реальна мережа з 5 об'єктів має менше ніж 9 попарних каналів зв'язку, то відсутні зв'язки представляються на графі G(5) ребрами з нульовою вагою (тобто, нульовою пропускною спроможністю умовних каналів зв'язку). Параметри графу G(V) як функції від числа вершин V обчислюються за формулами 7÷12 (рис. 3.5).

Структура таблиці Inv_task(8, 11) визначається параметрами графу G(5).

1) Параметр «Кількість ребер E(n)» обчислюється по формулі 8 (рис. 3.5):

$$PGE(V) = 3\cdot(V-2) = 3\cdot(5-2) = 9 \text{ ребер.}$$

Його значення обумовлює кількість стовпців масиву Inv_task(8, 11):

11 стовпців = (9+1+1):

дев'ять ребер E(n) кожного шляху +

+ один номер k шляху $P_k(n)$ +

+ одне ціле невід'ємне значення максимально можливого потоку f(k)

уздовж шляху $P_k(n)$ що дорівнює його мінімальному ребру min E(n).

2) Параметр «Кількість незалежних ST-шляхів» між полюсами S і T графу G(5) обчислюється по формулі 10 (рис. 3.5):

$$STP(V) = 1+ (V-1)\cdot(V-2)/2 = 1+ (5-1)\cdot(5-2)/2 = 6 \text{ шляхів.}$$

Його значення обумовлює кількість строк масиву Inv_task(8, 11):

8 строк = (6+1+1):

шість строк для 6 векторів незалежних шляхів P_k по 9 ребер +

+ один вектор E з дев'яти ребер E(n) графу G(5) +

+ один вектор RE(n) з 9 ребер остаточного графу ResG(5).

3) Параметр «Кількість варіантів дерева пошуку» ST-шляхів між полюсами S і T графу G(5) обчислюється по формулі 12 (рис. 3.5):

$$N(V) = (V-2)^2+1 = (5-2)^2+1 = 10 \text{ варіантів.}$$

Його значення обумовлює структуру і розмір модуля InverseTask. При цьому, дерево пошуку шляхів не є вхідними даними для програмного модуля Inv_task, оскільки воно є однаковим для всіх нормалізованих ST-графів G(V) для $V \le Vmax$. Тому, дерево пошуку шляхів можна створити один раз рекурсивним способом за допомогою штучного інтелекту для довільного нормалізованого графу G(V), $V \le Vmax$ (розділ 3.2 даної роботи).

За цих обставин, дерево пошуку шляхів можна реалізувати простим без-цикловим кодом з паралельним обчисленням на програмованих логічних інтегральних схемах ПЛІС (FPGA). У даному програмному модулі InverseTask, параметр Vmax, за визначенням, становить 5 вершин графу.

Розглянемо функціональні блоки модуля InverseTask (див. рис. 3.12).

Блок «Declare/Set»:

а) Декларує масиви операційних даних:

– таблицю Inv_task як двомірний кортеж даних розміром (8×11);

– вектор потоків по шляхам P_k як одномірний масив f(k) розміром 6;

– матрицю шляхів P(k, n) як двомірний масив розміром (6×9);

– вектор ребер RE(n) остаточного графу як одномірний масив ResG(9).

б) Задає вектор ребер графу E(n) як одномірний масив розміром 9.

Блок «Loop by n∈[1, 9]»

Після старту, копіює вхідний вектор ребер E(n) у вектор остаточного графу RE(n), який відображує поточний стан графу на кожному k-му кроці обчислення потоку f(k) на шляху P_k. Після розподілу усіх 6 потоків по шляхам P_k вектор RE(n) містить остаточні (надлишкові) значення ребер графу.

Блок «Flows» програмного модуля InverseTask (рис. 3.12) вирішує зворотну задачу обчислення максимального потоку і його розподілу по шляхах.

Шлях P_k вважається *незалежним*, якщо він містить *хоча б одне унікальне ребро*, відсутнє у будь якому іншому шляху P_m, m≠k. Якщо всі ребра шляху є унікальними, то він є *повністю незалежним* (безальтернативним).

Граф G(5) – див. рис. 3.6 – містить чотири повністю незалежні ST-шляхи між вершинами «1» і «2»:

– один найкоротший (1-hop) шлях по ребру 1;

– три 2-hop шляхи по ребрах: (2, 3), (5, 6), (8, 9).

Решта два (з шести можливих незалежних шляхів) є альтернативними, тобто мають спільні ребра з деякими іншими шляхами.

Обчислення потоків виконується у порядку зростання довжини шляху (по кількості ребер), починаючи з найкоротшого 1-hop шляху P1 (ребро 1).

Значення ребер беруться з масиву RE ребер остаточного графу ResG(5).

Блок ‘Flows” складається з п’яти сегментів для дерева шляхів:

P1; P2÷P4; P5(P51/P52); P5(P53/P54); P6(P61/P62). Ці сегменти містять 10 кроків обчислення дерева шляхів: P1, P2, P3, P4, P51, P52, P53, P54, P61, P62.

У блоці «Flows» програмного модуля InverseTask (див. рис. 3.12) перший безальтернативний (найкоротший) шлях P1 визначає потік f(1) по значенню ребра 1, тобто f(1) = RE(1). Після цього ребро 1 «насичується»: RE(1) = 0. Три інші безальтернативні шляхи P2÷P4 обробляються по однотипній схемі, наприклад, для шляху P2:

– у змінну P2 записуються значення ребер відповідного шляху:

P2 = [RE(2), RE(3)];

– мінімальне ребро шляху P2 записується у поточну змінну fm:

fm = min P2;

– призначається потік f(2) = fm;

– обидва ребра шляху P2 зменшують вагу на величину fm

(при цьому, щонайменше, одне з них насичується (обертається в нуль):

RE(2)=RE(2)–fm; RE(3)=RE(3)–fm.

Два останні (альтернативні) 3-hop шляхи P5, P6 обробляються по більш складній схемі. Як вказано вище, на графі G(5) може бути 6 незалежних шляхів, натомість, 10 варіантів на дереві пошуку; чотири варіанти і чотири потоки f(1)÷f(4) на шляхах P1÷P4 обчислені вище. Залишилось 6 варіантів перебору шляхів для обрання двох останніх потоків f(5), f(6) по три варіанти на кожний з двох останніх можливих незалежних шляхів P5 і P6.

Згідно формули 9 таблиці 6, кількість можливих 3-hop шляхів на графі G(V) дорівнює STPh(V, h) = V– h = 5–3= 2 шляхи; кількість 4-hop шляхів – відповідно V–h = 5–4 =1 шлях. На кожен з цих трьох шляхів (два 2-hop шляхи + один 4-hop шлях) існує по два варіанти побудови альтернативних шляхів. Загалом, далі треба ще обчислити 6 варіантів дерева пошуку шляхів.

В залежності від конкретних значень ребер E(n) графу G(5), може бути два наступні результати для потоків f(5), f(6).

1) Два 3-hop потоки f(5), f(6) по шляхах з множини {P51, P52, P53, P54}:
   f(5) по P51(5, 4, 3) / P52(2, 4, 6); f(6) по P53(8, 7, 6) / P54(5, 7, 1);
2) Один 3-hop потік f(5) по шляхах з множини {P51, P52, P53, P54} +
   + один 4-hop потік f(6) по шляхах з множини {P61, P62}.

Висновки до розділу 3

У даному розділі дисертаціної роботи розроблено комп'ютерний алгоритм для крипто-стійкого динамічного кодування в мобільній ad-hoc мережі Інтернету речей (IoT) на основі недетермінованої машини Тьюринга, теоретичні засади якого сформульовані у розділі 2. Побудовано блок-схему алгоритму і програмний модуль TMSend на мові Python. Наведено типові приклади симуляції недетермінованої машини Тьюринга NdTM у середовищі Python/Mac OS. Визначено три наступні базові профіля роботи роботи алгоритму.

а) «Гнучка структура» (без крипто-захисту): гнучке формування структури фрейму при відсутності кіберзагроз для зменшення службової інформації та підвищення інформаційної ефективності каналу зв'язку; дає виграш інформаційної ефективності у 1.5÷6.6 рази.

б) «Крипто-захист»: динамічне крипто-стійке кодування фрейму змінним ключом що синхронізується між взаємодіючими сторонами; забезпечує еквівалентний до існуючих методів («нормальний») рівень захисту простими засобами незалежно (або на додаток) до традиційних методів шифрування; забезпечує достатню для ad-hoc мережі стійкість до зламу і виграш інформаційної ефективності у 2.5÷9.6 рази (в залежності від розміру даних і типу інкапсуляції).

в) «Пост-шифрування»: додаткове крипто-кодування корисних та службових даних тимчасовим ключем, номер якого передається відкритим параметром і визначає значення ключа в секретному масиві, що наперед випадково згенеровано і завантажено в пам'ять взаємодіючих сторін; даний механізм реалізує підвищений проактивний рівень захисту даних на додаток до відомих методів шифрування і дозволяє підвищити рівень захисту відомих методів; підвищує рівень захисту незалежно і на додаток до існуючих методів.

Досліджено структурно-параметричні характеристики нормалізованого планарного графу ad-hoc мережі (ST-графу). Систематизовано і розширено базові формули комбінаторики з шістьма новими функціями для обчислення параметрів ST-графу за кількістю його вершин.

Розглянуто структуру шляхів ST-графу з точки зору побудови дерева пошуку найкоротших шляхів.

Удосконалено методи рішення комбінаторних задач на графах *шляхом* обґрунтування рекурсивної побудови дерева пошуку найкоротших шляхів на вільно-орієнтованому графі мережі та комбінаторної формули від числа вершин графу для визначення необхідної кількості ітерацій, *що дозволяє* прискорити алгоритми оптимізації потоків даних в мережах, а також залучити штучний інтелект до реалізації таких алгоритмів (**четвертий науковий результат**).

Розроблено комбінаторну модель оптимізації розподілу даних ad-hoc мережі Інтернету речей з використанням штучного інтелекту, а також комп'ютерний алгоритм динамічного розподілу даних в мережі IoT.

Імплементація цього алгоритму розширює можливості адаптації ad-hoc мережі Інтернету речей до динамічно змінюваних факторів для *максимально ефективного використання наявних ресурсів* (енерго-запаси мобільних об'єктів, канали зв'язку, мережеві інтерфейси тощо).

Це підсилює функціональну стійкість мережі, і зокрема, рівень кіберзахисту, що важливо при організації спеціальних місій для групи автономних мобільних роботизованих платформ, які об'єднуються в ad-hoc мережу змінної топології і метрики.

ВИСНОВКИ

Сукупність наукових результатів дисертаційного дослідження дозволяє розв’язати актуальну науково-прикладну задачу, що полягає у створенні проактивних методів криптостікого динамічного кодування і розподілу даних для підвищення захисту інформації в системах та мережах Інтернету речей.

Основним науковим результатом дисертаційної роботи є створення методу крипто-стійкого динамічного кодування і розподілу даних мобільної ad-hoc мережі Інтернету речей для підвищення захисту інформації. *На відміну від існуючих*, даний метод застосовує недетерміновану машину Тьюринга зі змінною у часі системою команд для гнучкого формування структури даних в каналі зв’язку, а також комбінаторну модель оптимального розподілу даних для рекурсивної побудови дерева пошуку найкоротших шляхів на вільно-орієнтованому планарному графі з використанням штучного інтелекту, *що дозволяє* підвищити проактивний кіберзахист мережі, спростити його апаратно-програмну реалізацію, підвищити ефективність каналів зв’язку, обмежити людський фактор у питаннях конфіденційності, збільшити швидкодію адаптивної реконфігурації розподілу даних та підсилити функціональну стійкість в умовах динамічно змінних факторів впливу.

У межах основного наукового результату роботи отримано наступне.

1) *Удосконалено* модель динамічного кодування даних мобільної ad-hoc мережі Інтернету речей *шляхом* розробки і застосування трирівневої структури інтерфейсів роботизованого мобільного комплексу та відповідної формальної граматики побудови фреймів в каналі зв’язку з трійковим лінійними сигналами для нестандартного утворення цифрового потоку з окремих незалежних функціональних елементів у довільному порядку і складі, *що дозволяє* збільшити ефективність каналу у 1.5÷6.6 разів та ускладнити його злам за рахунок скорочення службової інформації і непередбачуваного формату даних.

2) *Отримала подальший розвиток* методика застосування теорії автоматів шляхом розробки недетермінованої машини Тьюринга для структуризації даних в каналі зв’язку, *відмінна тим*, що через випадкові інтервали часу, кодові кортежі команд перемішуються і змінюють свою семантику у порядку відомому лише конкретному адресату, *що дозволяє* створювати різноманіття алгоритмів криптографічного захисту інформації незалежно від існуючих апаратно-програмних засобів.

3) *Вперше запропоновано* спосіб динамічної синхронізації профільного ключа крипто-стійкого кодування між бортовим контролером мобільного об’єкту та базовою станцією дистанційного управління на основі недетермінованої машини Тьюринга, *відмінний тим*, що на кожну місію мобільного об’єкту генерується і закладається в пам’ять обох сторін унікальна випадкова послідовність профільних ключів, а номер ключа у цій послідовності відіграє роль відкритого ключа для передачі по каналу зв’язку, *що дозволяє* підвищити проактивний кіберзахист мережі, спростити його апаратно-програмну реалізацію і підвищити у 2.5÷9.6 разів ефективність каналів зв’язку, а також обмежити людський фактор у питаннях конфіденційності.

4) *Удосконалено* методи рішення комбінаторних задач на графах *шляхом* обґрунтування рекурсивної побудови дерева пошуку найкоротших шляхів на вільно-орієнтованому графі мережі та відповідної комбінаторної формули від числа вершин графу для визначення необхідної кількості ітерацій, *що дозволяє* прискорити у 1.3÷1.5 рази алгоритми оптимізації потоків даних в мережах, а також залучити штучний інтелект до реалізації таких алгоритмів.

Результати дисертації сприяють підвищенню рівня кіберзахисту систем та мереж Інтернету речей в умовах зростання кіберзагроз та складності атак. Впровадження результатів дисертації дозволить підвищити стійкість та надійність функціонування критично важливих мереж, що є необхідною передумовою для безпечного розвитку цифрового суспільства.

Практично значущими є наступні результати роботи.

– Для суб'єктів критичної інфраструктури та промислових підприємств. Розроблена методологія крипто-стійкого динамічного кодування та розподілу даних може бути використана для забезпечення надійного та стійкого захисту від атак типу «відмова в обслуговуванні» (DoS), ін’єкцій шкідливого коду та несанкціонованого доступу. Це дозволить запобігти збою енергосистем типу Smart Grid, промислових IoT-мереж і транспортних систем.

– Для розробників та виробників IoT-пристроїв. Запропоновані алгоритми та протоколи, що використовують штучний інтелект для адаптивного керування кодуванням, можуть бути інтегровані у програмне забезпечення нових пристроїв. Це забезпечить проактивний захист, що дозволить системам автономно протистояти новим, раніше невідомим загрозам без необхідності втручання оператора.

– Для розробників та користувачів мобільних роботизованих об'єктів та ad-hoc мережам. Запропонована методологія забезпечує підвищену стійкість зв’язку та маскування трафіку в умовах активної радіоелектронної боротьби. Динамічний розподіл даних та адаптивне кодування дозволяють протидіяти навмисним перешкодам і запобігати втраті управління та інформації в роях дронів та інших автономних системах, що працюють у ворожому середовищі. Це є критично важливим для безпеки та ефективного виконання бойових, розвідувальних та рятувальних місій.

– Для наукових установ та закладів вищої освіти. Розроблена методологічна база слугує теоретичним підґрунтям для подальших наукових досліджень у сфері кібербезпеки децентралізованих систем. Результати роботи можуть бути використані для модернізації навчальних курсів та спеціалізованих дисциплін з інформаційної безпеки та штучного інтелекту.

Окремі результати роботи впроваджено на ТОВ «Радуга-N» для послуг «Розумний дім»» (спосіб динамічної синхронізації поточного профільного ключа між контролером МО і БС, Додаток Б), у навчальні дисципліни ДУІТЗ: «Інформаційні технології кодування і забезпечення завадостійкості систем» (методи теорії автоматів і недетермінованої машини Тьюринга для крипто-стійкого кодування), «Інтернет речей» (контекстно-визначена модель взаємодії відкритих систем для дослідження проблем кібербезпеки IoT), «Методи та системи штучного інтелекту» (алгоритм комбінаторної оптимізації розподілу даних з використанням штучного інтелект (Додаток В), а також у навчальний процес та наукову діяльність Національного університету «Одеська політехніка» (Додатки Г, Д).

Основні результати дисертаційної роботи представлені і обговорені на 7 міжнародних науково-практичних конференціях (НПК) та відображені у 11 наукових публікаціях, у тому числі, 4 статті у наукових фахових виданнях що індексуються у НМБД Scopus (у співавторстві), 7 опублікованих тез доповідей на міжнародних НПК (з них 5 одноосібних).

Дисертаційна робота підготовлена у відповідності до законів України «Про основні засади забезпечення кібербезпеки України» від 27.03.2025, «Про захист інформації в інформаційно-комунікаційних системах» від 20.04.2025, «Про Стратегію кібербезпеки України» (рішення Ради національної безпеки та оборони України від 14.05.2021). Тема роботи корелюється з пунктом 1.2.9.5 («Дослідження та розроблення методів інформаційної безпеки комп'ютерних систем і мереж») положень Національної академії наук України «Основні наукові напрями та найважливіші проблеми фундаментальних досліджень у галузі природничих, технічних і гуманітарних наук Національної академії наук України на 2024–2028 роки» No8 від 10.01.2024.

Тема дисератційної роботи відповідає науковим планам і тематиці досліджень, а також програмам підготовки студентів Національного університету «Одеська політехніка» у сфері кібербезпеки, інтелектуальних систем та робототехніки. Дисертація є складовою частиною НДР No 199-64 «Моделі, методи розробки та апаратні рішення високочастотних компонентів програмованих мобільних систем» Інституту штучного інтелекту та робототехніки Національного університету «Одеська політехніка» (No держреєстрації 0121U110245).

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України «Про основні засади забезпечення кібербезпеки України» від 27.03.2025.
2. Закон України «Про захист інформації в інформаційно-комунікаційних системах» від 20.04.2025.
3. Закон України «Про Стратегію кібербезпеки України» (рішення Ради національної безпеки та оборони України від 14.05.2021.
4. Постанова Президії НАНУ від 10.01.2024 No8 «Основні наукові напрями та найважливіші проблеми фундаментальних досліджень у галузі природничих, технічних і гуманітарних наук НАН України на 2024-2028».
5. Постанова КМ України № 942 від 7.09.2011 «Про затвердження переліку пріоритетних тематичних напрямів наукових досліджень і науково-технічних розробок на період до 2023 року».
6. Постанова КМ No 463 від 9.05.2023 про зміни до закону «Про затвердження переліку пріоритетних тематичних напрямів наукових досліджень і науково-технічних розробок на період до 2023 року».
7. Національний університет «Одеська політехніка». Інститут штучного інтелекту та робототехніки. URL: https://op.edu.ua/iair/activity.
8. Тіхонов С.В. Захищене кодування мережевих інтерфейсів у розподілених системах інтернету речей на основі багаторівневих формальних граматик. Proc. of the VII International Scientific and Practical Conference «Science and Technology: Problems, Prospects and Innovations» (Osaka, 13-15 April, 2023). С. 174-180. URL: https://sci-conf.com.ua/wp-content/uploads/2023/04/SCIENCE-AND-TECHNOLOGY-PROBLEMS-PROSPECTS-AND-INNOVATIONS-13-15.04.23.pdf.
9. Tikhonov Serhii. Hierarchical data coding approach based on adaptive formal grammar. *Proc. of the XIV International Scientific and Practical Conference "Science and Practice: Implementation to Modern Society"* (Manchester, 26-28 April, 2023), № 152. P. 559-563. URL: https://archive.interconf.center/plugins/generic/pdfJsViewer/pdf.js/web/viewer.html.2C1040%2C899%5D.

10. Ситніков В.С., Тіхонов С.В. Принципи захисту інформації у комп'ютерних системах Інтернету речей на основі ієрархічного кодування у формальних граматиках. *Збірник тез 24-ї Міжнародної НПК «Сучасні інформаційні та електронні технології» CIET/MIET-2023* (Одеса, 29-31 травня 2023). С. 31-32. URL: https://old.tkea.com.ua/siet/archive/2023/31.pdf.
11. Тіхонов С.В. Формальна граматика машини Тьюринга для завадостійкого кодування фреймів Ethernet на основі трійкових лінійних сигналів. *Proc. of XII Internetional Scientific and Practical Conference «Innovations and Prospects in Modern Science»* (Stockholm, 20-22 November, 2023). С. 315-325. URL: https://sci-conf.com.ua/wp-content/uploads/2023/11/INNOVATIONS-AND-PROSPECTS-IN-MODERN-SCIENCE-20-22.11.23.pdf.
12. Тіхонов С. Питання кібербезпеки в базових технологіях інтернету речей. *Proc. of XII International Scientific and Practical Conference "Current Challenges of Science and Education"* (Berlin, 29-31 July, 2024). С. 165-171. URL: https://sci-conf.com.ua/wp-content/uploads/2024/07/CURRENT-CHALLENGES-OF-SCIENCE-AND-EDUCATION-29-31.07.24.pdf.
13. Тіхонов В., Тіхонов С., Яворська О. Дослідження комбінаторних потокових задач на планарному графі комп'ютерної мережі. *Proc. of XI International Scientific and Practical Conf. "Science in the Modern World: Innovations and Challenges"* (Toronto, 10-12 July 2025). С. 45-52. URL: https://sci-conf.com.ua/wp-content/uploads/2025/07/SCIENCE-IN-THE-MODERN-WORLD-INNOVATIONS-AND-CHALLENGES-12-14.06.2025.pdf.
14. Тіхонов С.В. Підвищення стійкості ad-hoc мережі IoT на основі швидкого алгоритму розподілу даних з використанням ШІ. *Proc. of V International Scientific and Practical Conference "International Experience in Scientific Rresearch"* (Chicago, 18-20 December 2025). P. 300-308. URL: https://sci-conf.com.ua/wp-content/uploads/2025/12/INTERNATIONAL-EXPERIENCE-IN-SCIENTIFIC-RESEARCH-18-20.12.25.pdf.
15. Tikhonov V., Taher A., Tikhonov S., Shulakova K., Hluschenko V., Chaika A. Turing Machine Development for High-secure Data Link Encoding in the Internet

of Things Channel. *Hochschule Anhalt Edition "Applied Innovations in Information and Communication Technology" (ICAIIT)*. 2024. V. 12, № 1. P. 1-10. ISSN 2199-8876. DOI: http://dx.doi.org/10.25673/115635.

16. Tikhonov V., Siemens E., Vasiliu Y., Sitnikov V., Taher A., Tykhonova O., Shulakova K., Tikhonov S. Context-Defined Model of Open Systems Interaction for IoT Cybersecurity Issues Study. *Hochschule Anhalt Edition "Proceedings of the International Conference on Applied Innovations in IT" (ICAIIT)*. 2024. V. 12, № 2, P. 35–44. ISSN 2199-8876. DOI: http://dx.doi.org/10.25673/118112.
17. Tikhonov V., Taher A., Shulakova K., Tikhonov S., Demchenko O. Turing Machine Development for High Protected Remote Control of the IoT Mobile Platform. *Springer Nature Link, Lecture Notes in Networks and Systems, ICAIIT-2024*. 2025. V. 1338. P. 61–80. ISSN 2367-3370, 2367-3389 (electronic). DOI: https://doi.org/10.1007/978-3-031-89296-7_5.
18. Tikhonov V., Sitnikov V., Tikhonov S. An Improved AI-Driven Algorithm of Data Flow Optimization on the Normalized Bipolar Planar Free-Oriented Network Graph. *Hochschule Anhalt Edition "Proceedings of the International Conference on Applied Innovations in IT" (ICAIIT)*. 2025. V. 13, № 5. P. 7–14. ISSN 2199-8876. DOI: http://dx.doi.org/10.25673/122800.
19. Neumann J. Die Axiomatisierung der Mengenlehre. *Mathematische Zeitschrift.* 1928. V. 27, P. 669-752. URL: https://gdz.sub.uni-goettingen.de (visited: 03.01.2026).
20. Kron G. *Diakoptics: Piecewise Solution of Large-Scale Systems*. London, Macdonald & Co., Ltd., 1963. 166 p. URL: https://www.semanticscholar.org/paper/Diakoptics%3A-The-piecewise-solution-of-large-scale-9-Ku/58ffda9ac9a51887d0 (visited: 03.01.2026).
21. Mesarovic M.D. *General Systems Theory and its Mathematical Foundation*. System research center, Case Western Reserve University, 1967. 22 p. URL: https://apps.dtic.mil/sti/tr/pdf/AD0659485.pdf (visited: 03.01.2026).
22. Von Bertalanffy L. *General Systems Theory. Foundations, Development, Applications*. University of Alberta Edmonton, Canada, George Braziller, new

York, 1968. 289 p. URL: https://www.google.com/url?esrc=s&q=&rct=j&sa=U&url=https://tomek.org/files/general_system_theory_foundations_development_applications.pdf&ved=2ahUKEwjcwcX_aUAxVPBNsEHVrmK3cQFnoECAgQAg&usg=AOvVaw1gUGwR4ZEVe715L_v-puRa (visited: 03.01.2026).

23. Abomhara M., Koien G.M., Alghamdi M. Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks. *Journal of Cyber Security*, 2015. Vol. 4. P. 65–88. URL: https://www.researchgate.net/publication/349642687_Cyber_Security_and_the_Internet_of_Things (visited: 03.01.2026).
24. PROFIBUS System Description Technology and Application. URL: https://www.profibus.com/fileadmin/media/downloadsection/PROFIBUS_Systembeschreibung_ENG_web.pdf (visited: 03.01.2026).
25. Feldbus – M-Bus & M-Bus Wireless. URL: https://www.wachendorff-prozesstechnik.de/technologie/feldbus/m-bus/ (visited: 03.01.2026).
26. Payne S. Zigbee Technology: Everything You Need to Know for IoT. URL: https://innexia.in/zigbee-iot-guide/ (visited: 03.01.2026).
27. ITU-T: Y.2060. Overview of the Internet of things. URL: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-Y.2060-201206-I!!PDF-E&type=items (visited: 03.01.2026).
28. IoT Reference Model Introduced at IoT World Forum 2014. URL: https://ytd2525.wordpress.com/2014/10/22/iot-reference-model-introduced-at-iot-world-forum-2014/ (visited: 03.01.2026).
29. IoT World Forum Reference Model. URL: https://www.researchgate.net/figure/oT-World-Forum-Reference-Model_fig2_323525875#:~:World%20Forum%20(IWF,%2Dbased%20IoT.%20... (visited: 03.01.2026).
30. What is Industry 4.0. URL: https://www.everythingrf.com/community/what-is-industry-4-0?gad_source=1&gad_campaignid=197404533&gbraid=0AAAADxGRBPI86jxBincgn5rF-zbaHcBd&gclid=CjwKCAiAoNbIBhB5EiwAZFbYGBbnTUSJ5_j_pW06v7IYYZjNHIuOBavKtOjD3jnpYlBXKxcOuMXaxhoCz6gQAvD_BwE (visited: 03.01.2026).

31. Jeffrey Voas. NIST Special Publication 800-183. Networks of ‘Things’. URL: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-183.pdf (visited: 03.01.2026).
32. Dotdot was a huge hit at CES 2017. URL: https://www. symmetryelectronics.com/blog/dotdot-was-a-huge-hit-at-ces-2017/ (visited: 03.01.2026).
33. What is Dotdot? URL: https://en.dsr-corporation.com/news/ what-is-dotdot.
34. IEEE 802.15.4-2020: IEEE Standard for Low-Rate Wireless Networks. URL: https://standards.ieee.org/ieee/802.15.4/7029/ (visited: 03.01.2026).
35. Industrial Internet Consortium: Global Industry Standards for Industrial IoT. URL: https://www.iiconsortium.org/pdf/IIC_Global_Standards_Strategy_ Whitepaper.pdf (visited: 03.01.2026).
36. Industry IoT Consortium. The Industrial Internet of Things Connectivity Framework. URL: https://www.iiconsortium.org/iicf/ (visited: 03.01.2026).
37. Razzaq M.A. et al. Security Issues in the Internet of Things (IoT): A Comprehensive Study. *International Journal of Advanced Computer Science and Applications*. 2017. V. 8, № 6. P. 383-388.
38. Industry IoT Consortium. The Industrial Internet Reference Architecture (IIC-IIRA v.1.10, 2022). URL: https://www.iiconsortium.org/wp-content/uploads/ sites/2/2022/11/IIRA-v1.10.pdf (visited: 03.01.2026).
39. Industrial Internet Consortium. Industrial Internet of Things. Volume G4: Security Framework (IIC-IISF 2016). URL: https://www.iiconsortium.org/pdf/IIC_ PUB_G4_V1.00_PB.pdf (visited: 03.01.2026).
40. Teixeira de Sousa P., Stuckmann P. *Telecommunication Systems and Technologies, V. II - Telecommunication network interoperability.* 2009. 422 p. URL: http://eolss.net/sample-chapters/c05/E6-108-22.pdf (visited: 03.01.2026).
41. Zigbee Security 101 (Architecture and Security Issues, 2023). URL: https://payatu.com/blog/zigbee-security101architecture-and-security-issues/#:~: text=ZigBee%20standard%20permits%20the%20rewants%20to%20rejoin%20 the%20network (visited: 03.01.2026).

42. Wang J., Li Z., Sun M., Lui J.C.S. Zigbee's Network Rejoin Procedure for IoT Systems: Vulnerabilities and Implications. *25-th International Symposium on Research in Attacks, Intrusions and Defenses* (Limassol, 26-28 October, 2022), P. 292-307. DOI: https://doi.org/10.1145/3545948.3545953.
43. Клименко В.В. *Розробка безпечної бездротової мережі для керування елементами розумного будинку*: дисертація бакалавра: спец. 125-Кібербезпека / НУ «Одеська політехніка», Одеса, 2022. 48 с.
44. Kotenko I., Izrailov K., Buinevich M. Static Analysis of Information Systems for IoT Cyber Security: A Survey of Machine Learning Approaches. *Sensors*. 2022, V 22, № 4. P. 13-35. DOI: https://doi.org/10.3390/s22041335.
45. Washington Univ. St. Louis. Office of Information Security. Information Security for Researchers. The CIA Triad: Confidentiality, Integrity, and Availability. URL: https://informationsecurity.wustl.edu/information-security-for-researchers/ (visited: 03.01.2026).
46. Тріада інформаційної безпеки: Конфіденційність, цілісність, доступність (ЦРУ). URL: https://ukrayinska.libretexts.org/Робоча_сила/Комп'ютерні_програми_та_інформаційні_технології/ _(ЦРУ) (visited: 03.01.2026).
47. From WEP to WPA3. A brief history of Wi-Fi security protocols. URL: https://arstechnica.com/gadgets/2019/03/802-eleventy-who-goes-there-wpa3-wi-fi-security-and-what-came-before-it/ (visited: 03.01.2026).
48. Денбновецький С.В., Мельник І.В., Писаренко Л.Д. *Кодування сигналів в електронних системах. Ч. 3. Способи кодування сигналів. Том 1. Натуральні, ефективні та лінійні коди.* Київ: КПІ ім. І. Сікорського, 2021, 470 с.
49. Денбновецький С.В., Мельник І.В., Писаренко Л.Д. *Кодування сигналів в електронних системах. Ч. 3. Способи кодування сигналів. Том 2. Групові, ітаративні та згорткові коди.* Київ КПІ ім. І. Сікорського, 2021, 634 с.
50. Денбновецький С.В., Мельник І.В., Писаренко Л.Д. *Кодування сигналів в електронних системах. Ч. 3. Способи кодування сигналів. Том 3. Методи стиснення числової та текстової інформації.* Київ, КПІ ім. І. Сікорського, 2022, 690 с.

51. Йона Л.Г., Байрамова Ю.М. *Методи побудови криптографічних систем*. Міжнародний гуманітарний університет, 2024, 32 с.
52. Federal Information Processing Standards Publication: FIPS 197. Advanced Encryption Standard (AES, 2001/Uptd. 2023). URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf (visited: 03.01.2026).
53. Rivest R., Shamir A., Adleman L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Massachusetts Institute of Technology, Scientific American Magazine*. 1977. V. 237, №5. P. 1-15. URL: https://people.csail.mit.edu/rivest/Rsapaper.pdf (visited: 03.01.2026).
54. Hankerson D., Menezes A., Vanstone S. *Guide to Elliptic Curve Cryptography*. Springer, 2004, 332 p. URL: https://theswissbay.ch/pdf/Gentoomen%20Library/Cryptography/Guide%20to%20Elliptic%20Curve%20Cryptography%20-%20Darrel%20Hankerson.pdf.
55. Rivest R., Shuldt J.C. Spritz – A Spongy RC4-like Stream Cipher. URL: http://crypto.2014.rump.cr.yp.to/6dd60fdcd345b5b0486c017959fab15b.pdf (visited: 03.01.2026).
56. Data Encryption Standard (DES). URL: https://nvlpubs.nist.gov/nistpubs/sp958-lide/250-253.pdf (visited: 03.01.2026).
57. SSL 3.0 Protocol Vulnerability and POODLE Attack. URL: https://www.cisa.gov/news-events/alerts/2014/10/17/ssl-30-protocol-vulnerability-and-poodle-attack (visited: 03.01.2026).
58. Soni A., Upadhyay R, Jain A. Internet of Things and Wireless Physical Layer Security: A Survey. *In book: "Computer Communication, Networking and Internet Security"*. 2017. P.115-123. DOI: 10.1007/978-981-10-3226-4_11. URL: https://www.researchgate.net/publication/316713496_Internet_of_Things_and_Wireless_Physical_Layer_Security_A_Survey?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6Il9kaXJlY3QiLCJwYWdlIjoiX2RpcmVjdCJ9fQ.
59. Lehembre G. Wi-Fi security – WEP, WPA and WPA2. URL: https://repository.root-me.org/Réseau/EN%20-%20Hacking%20wifi.pdf (visited: 03.01.2026).

60. Чи можна зламати AES-256? URL: https://night.kozak.cx.ua/ukraincyam/chi-mozhna-zlamati-aes-256.html (visited: 03.01.2026).
61. What Is WPA3? URL: https://jumpcloud.com/it-index/what-is-wpa3#:~:text=WPA3%20is%20a%20Wi%2DFi,a%20more%20secure%20wireless%20experience (visited: 03.01.2026).
62. Кузьменко І.М. Теорія графів. *Теорія графів. Навч. посіб. для бакалаврів за спец. 122 Комп'ютерні науки*. Київ: НУ КПІ ім. І. Сікорського, 2020. 71 с.
63. Tikhonov V., Nesterenko S., et al. Inverse and Direct Maxflow Problem Study on the Free-Oriented ST-Planar Network Graph. *Hochschule Anhalt Edition "Proceedings of the International Conference on Applied Innovations in IT" (ICAIIT)*. 2023. V. 11, № 2, P. 1-10. DOI: 10.25673/112988.
64. Ford Jr. L.R., Fulkerson D.R. Maximal Flow Through a Network. *Canadian Journal of Mathematics*. 1956. P. 399-404. URL: https://www.semanticscholar.org/venue?name=Canadian%20Journal%20of%20Mathematics (visited: 03.01.2026).
65. Dinitz Y. Algorithm for Solution of a Problem of Maximum Flow in a Network with Power Estimation. 1970. URL: https://www.researchgate.net/publication/228057696_Algorithm_for_Solution_of_a_Problem_of_Maximum_Flow_in_Networks_with_Power_Estimation (visited: 03.01.2026).
66. Edmonds J., Karp R. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *Journ. of Ass. for Computing Machinery*. 1972. V.19, №2. P. 248-264. URL: https://web.eecs.umich.edu/~pettie/matching/Edmonds-Karp-network-flow.pdf (visited: 03.01.2026).
67. Goldberg A.V., Tarjan R.E. A New Approach to the Maximum Flow Problem. *Proc. of the 18-th Annual ACM Symposium on Theory of Computing* (Berkeley, Calif., 28-30 May, 1986). P. 136-146. URL: https://www.cs.cmu.edu/afs/cs/academic/class/15499-s09/www/handouts/maxflow.pdf (visited: 03.01.2026).
68. Goldberg A. V., Rao S. Beyond the flow decomposition barrier. ACM Vol. 45(5), 1998, pages 783–797. DOI: https://doi.org/10.1145/290179.290181.

69. Papp D. Goldberg-Rao Algorithm for the Maximum Flow Problem. URL: https://www.cs.princeton.edu/courses/archive/fall06/cos528/handouts/Goldberg-Rao.pdf (visited: 03.01.2026).
70. Christiano P., Kelner J.A., et al. Electrical Flows, Laplacian Systems, and Faster Approximation of Maximum Flow in Undirected Graphs. *ArXiv Computer Science, Data Structures and Algorithms*. 2010. 34 p. DOI: https://doi.org/10.48550/arXiv.1010.2921.
71. Bondy J.A., Murty U.S.R. *Graph theory with applications*. Macmillan Press Ltd, GB, 1976. 264 p. URL: https://www.iro.umontreal.ca/~hahn/ift3545/gtwa.pdf (visited: 03.01.2026).
72. Williamson D.P. *Network Flow Algorithms*. Cornell University, 2019. 265 p. URL: https:// www.networkflowalgs.com/book.pdf (visited: 03.01.2026).
73. Tykhonova O., Yavorska O., Berezovskiy V. The Max-Flow Problem Statement on the Three-Pole Open Network Graph. *The 3rd Int. Conf. on "Advanced Information and Communications Technologies"* (Lviv, 2-6 July, 2019), P. 209-212.
74. Тихонова О.В. *Конвеєрно-модульний метод інтеграції мультимедійних потоків з контролем затримок в пакетних телекомунікаційних мережах*. Дисертація кандидата наук: 05.12.02 / ОНАЗ, Одеса, 2019. 161 с.
75. Haese R., Heller T., Krumke S.O. Algorithms and Complexity for the Almost Equal Maximum Flow Problem. *Operations Research Proceedings, 2019. Springer*, Cham, 2020. P. 323-329. DOI: https://doi.org/10.1007/978-3-030-48439-2_39.
76. Alzaben N., Engels D.W. End-to-End Routing in SDN Controllers Using Max-Flow Min-Cut Route Selection Algorithm. *Int. Conf. on Advanced Communication Technologies* (ICACT, 7-10 February 2021). P. 461-467.
77. Liu H., et al. Optimization of a Logistics Transportation Network Based on a Genetic Algorithm. *Hindawi Mobile Information Systems*, 2022. P. 167-176. DOI: https://doi.org/10.1155/2022/1271488.

78. Cruz-Mejía O., Letchford A.N. A survey on exact algorithms for the maximum flow and minimum-cost flow problems. *Networks*. 2023 V. 82, № 2. P. 167-176. DOI: https://doi.org/10.1002/net.22169.
79. Bengio Y., et al. Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research, Amsterdam*. 2021. V. 290, № 2. P. 405-421. DOI: 10.1016/j.ejor.2020.07.063.
80. Mor B., Shabtay D., Yedidsion L. Heuristic algorithms for solving a set of NP-hard single-machine scheduling problems with resource-dependent processing times. *Elsevier*. *Computers & Industrial Engineering*. 2021. V. 153. DOI: https://doi.org/10.1016/j.cie.2020.107024.
81. Jiang W. Graph-based Deep Learning for Communication Networks: A Survey. *Computer Communications*. 2022. V. 185. P. 40-54. URL: https://www.sciencedirect.com/science/article/abs/pii/S0140366421004874.
82. Baycik O. Machine learning based approaches to solve the maximum flow network interdiction problem. *Computers & Industrial Engineering*. 2022. V. 167. DOI: https://doi.org/10.1016/j.cie.2021.107873.
83. Tawanda T., et al. An intelligent node labelling maximum flow algorithm. *Int. Journal of System Assurance Engineering and Management*. 2023. V. 14. P. 1276-1284. DOI: https://doi.org/10.1007/s13198-023-01930-3.
84. Fong T., Nourbakhsh I., Dautenhahn K. A survey of socially interactive robots. *Robotics and Autonomous Systems*. 2003. №42. P. 143–166. DOI: 10.1016/S0921-8890(02)00372-X.
85. Klose R., Linz A., Ruckelshausen A. Robotic Platform for Remote Control and Teleservice in Agricultural Environments. *University of Applied Sciences Osnabrück, Faculty of Engineering and Computer Science*, Germany. URL: https://www.hs-osnabrueck.de/fileadmin/HSOS/Homepages/COALA/Veroeffentlichungen/2006-Bonn-Service.pdf (visited: 03.01.2026).
86. Haehnel H. Remote controlled flying robot platform. *Third Int. Conf. on Digital Information Management,* (London, 13-16 November 2008). P. 920-921. DOI: 10.1109/ICDIM.2008.4746850.

87. Laird R.T. Evolving U.S. Department of Defense (DoD). Unmanned Systems. Research, Development, Test, Acquisition & Evaluation (RDTA&E). *SPIE Proc. 7332: Unmanned Systems Technology XI, Defense Security Symposium*. (Orlando, 13-17 April 2009). DOI: 10.1117/12.832260

88. Tompa V., Hurgoiu D., et al. Remote control and monitoring of an autonomous mobile robot. *Proc. of 2012 IEEE Int. Conf. on Automation, Quality and Testing, Robotics* (Cluj-Napoca Romania24-27 May 2012). V. 1. P. 353-358. DOI: 10.1109/AQTR.2012.6237750.

89. Springer P.J. *Military robots and drones : a reference handbook*. Santa Barbara, Calif.: ABC-CLIO, 2013. 297 p. URL: https://nps.primo.exlibrisgroup.com/discovery/fulldisplay/alma991002802959703791/01NPS_INST:01NPS (visited: 03.01.2026).

90. Sathiyanarayan M. et al. Command controlled robot for military purpose. *International Journal for Technological Research in Engineering*. 2014. V. 1, № 9. P. 2347-4718. URL: https://www.researchgate.net/profile/Mithileysh-Sathiyanarayanan/publication/264120965_COMMAND_CONTROLLED_ROBOT_FOR_MILITARY_PURPOSE/links/53ced97c0cf2f7e53cf7e204.pdf (visited: 03.01.2026).

91. Devi A. Robot control using mobile phone. *Int. Journal of IEEE*. 2015. V. 7, № 2, P. 61-68. URL: https://www.semanticscholar.org/paper/ROBOT-CONTROL-USING-MOBILE-PHONE-Devi/c6f429bc1527e7118ece540a479ad8c1f688e78 (visited: 03.01.2026).

92. Almali M.N., et al. Wireless Remote Control of a Mobile Robot. *Int. Journal of Scientific Research in Information Systems and Engineering (IJSRISE*). 2015. Vol. 1, № 2. URL: https://www.researchgate.net/publication/303341093_Wireless_Remote_Control_of_a_Mobile_Robot (visited: 03.01.2026).

93. Duong P.M., et al. A novel platform for Internet-based mobile robot systems. 2016. URL: https://arxiv.org/pdf/1611.09433 (visited: 03.01.2026).

94. Karakaya S. et al. Virtual-reality-based remote-controlled mobile robot platform. *Global Journal of Computer Sciences Theory and Research*. 2017, V.7, №3. DOI:10.18844/gjcs.v7i3.2799.
95. Guizzo E. *How to get started in robotics*. URL: 2018 https://robotsguide. com/learn/how-to-get-started-in-robotics (visited: 03.01.2026).
96. Yamamoto T. et al. Development of Human Support Robot as the Research Platform of a Domestic Mobile Manipulator. *ROBOMECH Journal*. 2019. V. 6, № 1. P. 1-15. DOI: https://doi.org/10.1186/s40648-019-0132-3.
97. Rubio F. et al. Int. Journal of Advanced Robotic Systems, 2019. Vol. 16, № 2. P. 1-19. DOI: https://doi.org/10.1177/1729881419839596.
98. Romanov A.M. A review on control systems hardware and software for robots of various scale and purpose. Part 3. Extreme robotics. *RTJ*. 2020. V 8, № 3. P. 14-32. DOI: 10.32362/2500-316X-2020-8-3-14-32.
99. What is ArduPilot? URL: https://ardupilot.org (visited: 03.01.2026).
100. What is an AUV? URL: https://oceanexplorer.noaa.gov/ocean-fact/auv/ (visited: 03.01.2026).
101. Tsiu L., Markus E.D. A Survey of Formation Control for Multiple Mobile Robotic Systems. *Int. Journal of Mechanical Engineering and Robotics Research*. 2020, Vol. 9, № 11. P. 1515-1520. DOI: 10.18178/ijmerr.9.11.1515-1520.
102. Moniruzzaman M. et al. Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey. Robotics and Autonomous Systems . 2022. V. 150. P. 1-16. DOI: 10.1016/j.robot.2021.103973.
103. Maset E. et al. Performance Investigation and Repeatability Assessment of a Mobile Robotic System for 3D Mapping. *Robotics*. 2022. Vol.11, № 3. P. 1-16. DOI: 10.3390/robotics11030054.
104. Ikpeze O. et al. Smartphone Control Mobile Robot for Education and Research. Journal of Robotics. 2022. V.1. P.1-10. DOI: 10.1155/2022/5178629.

105. Sufiyan D. et al. An Efficient Multimodal Nature-Inspired Unmanned Aerial Vehicle Capable of Agile Maneuvers. *Advanced Intelligent Systems*. 2022. V. 5, № 1. P. 2200242-2200242. DOI: https://doi.org/10.1002/aisy.202200242.
106. Tsmots I. et al. Intelligent Motion Control System for the Mobile Robotic Platform . *In Proc. of the 3rd International Workshop on Computational Mechanics, Modern Mathematical Methods and Numerical Algorithms* (Kharkiv, 20-21 April, 2023). V. 3403. P. 418-433.
107. Stefanuto B. et al. Remote Lab of Robotic Manipulators Through an Open Access ROS-Based Platform. *IEEE 21-th Int. Conf. on Industrial Informatics* (Lemgo Germany, 17-20 July, 2023). P. 579–584. DOI: 10.1109/INDIN51400.2023.10218202.
108. Md Moniruzzaman et al. Structure-aware image translation-based long future prediction for enhancement of ground robotic vehicle teleoperation. *Advanced Intelligent Systems*, *Wiley-VCH GmbH*. 2023. V.5, № 10. P. 1-13. DOI: https://doi.org/10.1002/aisy.202200439.
109. Rubies E. et al. Remote Control Device to Drive the Arm Gestures of an Assistant Humanoid Robot. *Applied Sciences*. 2023. V. 13, № 19. DOI: 10.3390/app131911115.
110. Jiang T. et al. Development and Verification of an Autonomous and Controllable Mobile Robot Platform. *Mechatronincs and Intelligent Transportation Systems*. 2023, Vol. 2, № 1. P. 11-19. DOI: https://doi.org/10.56578/mits020102.
111. Rybczak M. et al. A survey of machine learning approaches for mobile robot control. *Robotics*. 2024. Vol. 13, № 1. P. 1-16. DOI: https://doi.org/10.3390/robotics13010012.
112. Zhang S., Zhao H. Research and implementation of cooperative control for ROS mobile robot. *Int. Conf. on Artificial Life and Robotics* (Oita Japan, 22-25 February, 2024). P. 43-48.
113. ISCER '24: Proceedings of the 2024 3rd International Symposium on Control Engineering and Robotics (China, 24-26 Mai, 2024). URL: https://dl.acm.org/doi/proceedings/10.1145/3679409?tocHeading=heading3.

114. Raz A. et al. Explainable AI and Robustness based Test and Evaluation of Reinforcement Learning. *IEEE Transactions on Aerospace and Electronic Systems*. 2024, P. 1-14. DOI: 10.1109/TAES.2024.3403078.
115. Geles I. et al. Demonstrating agile flight from pixels without state estimation. *Robotics: Science and Systems* (Delft Univ. of Technology, netherlend, 15-19 July, 2024). Session 8. Perception and navigation. Poster paper ID 82. DOI: https://doi.org/10.15607/rss.2024.xx.082.
116. Distributed Antenna Systems (DAS) Competency requirements. URL: https://www.etai.org/comps/DASI_comps.pdf (visited: 03.01.2026).
117. 802.11 Frame Types and Formats. URL: https://howiwifi.com/2020/07/13/802-11-frame-types-and-formats/ (visited: 03.01.2026).
118. Gast M.S. *802.11 Wireless Networks: The Definitive Guide. 2nd Ed. Ch. 4. 802.11 Framing in Detail*. 2005. 670 p.
119. IEEE P802.11. Wireless LANs. QoS Baseline Proposal – rev. 0. *Doc.: IEEE 802.11-00/360*. November 6, 2000. 60 p.
120. IEEE 802.11 Architecture (Updated 23 July, 2025). URL: https://www.geeksforgeeks.org/computer-organization-architecture/ieee-802-11-architecture/ (visited: 03.01.2026).
121. 802.11ac In-Depth // Aruba Networks. White paper. URL: https://higherlogicdownload.s3.amazonaws.com/HPE/MigratedAssets/WP_80211acInDepth.pdf (visited: 03.01.2026).
122. Jain N. PAM & Ethernet: A Perfect Match (2016). URL: https://www.edn.com/pam-ethernet-a-perfect-match/#google_vignette (visited: 03.01.2026).
123. 2.5GBASE-T and 5GBASE-T. URL: https://grokipedia.com/page/2.5GBASE-T_and_5GBASE-T (visited: 03.01.2026).
124. Гавриленко С.Ю. *Формальні мови, граматики та автомати. Навч. посібник*. Харків: НТУ «Харківський політехнічний інститут», 2021. 133 с.
125. Turing machines. URL: https://www.cl.cam.ac.uk/teaching/1718/CompTheory/lecture-7.pdf (visited: 03.01.2026).

126. Undecidable Problems for Turing Machines (Redux). *CSCI 541: Theory of Computing. Lecture 5*. 2021. P. 3-9. URL: https://people.stfx.ca/tjsmith/lec/F21CSCI541/F21CSCI541LecReducibilityPart2.pdf (visited: 03.01.2026).
127. Turing A. M. On computable numbers, with an application to the Entschidungs problem (1936). *Web archive*. P. 230-265. URL: https://www.astro.puc.cl/~rparra/tools/PAPERS/turing_1936.pdf visited: 03.01.2026).
128. CSCI 340: Computational Models. Post Machines. URL: https://assets.ctfassets.net/kdr3qnns3kvk/14dgLTw1gYAIEz86grXJwq/ae5aec6de2838732f80c5d5a283d351b/20-Post-Machines.pdf (visited: 03.01.2026).
129. Mealy G. H. A Method to Synthesizing Sequential Circuits. *Bell Systems Technical Journal*. 1955. V.34. P. 1045–1079. DOI: https://doi.org/10.1002/j.1538-7305.1955.tb03788.xDigital Object Identifier.
130. What is a state machine? URL: https://www.itemis.com/en/products/itemis-create/documentation/user-guide/overview_what_are_state_machines (visited: 03.01.2026).
131. Буряк Г.І. Елементи комбінаторики. *Коледж Ракетно-космічного машинобудування Дніпровського НУ ім. О. Гончара*. 2017. URL: http://dkrkm.org.ua/NMK/Buryak/e_konspekt_02.pdf (visited: 03.01.2026).
132. Тіхонов В. І., Тихонова О. В., Яворська О. М., Радкевич С. Д. Постановка задачі про максимальний потік на двополюсному відкритому вільно-орієнтованому графі телекомунікаційної транспортної мережі. *Вимірювальна та обчислювальна техніка в технологічних процесах: матеріали XX міжнар. наук.-техн. конф*. Одеса, 2020. С. 81–82.
133. Tikhonov V., Nesterenko S., Taher A., Tykhonova O., Tsyra O., Yavorska O., Shulakova K. Inverse and direct Maxflow problem study on the free-oriented ST-planar network graph. *Proc. of the International Conference on Applied Innovation in IT*. 2023. V. 11, No. 2. P. 1–10. DOI: 10.25673/112988.
134. Field Programmable Gate Arrays. URL: https://www.sciencedirect.com/topics/engineering/field-programmable-gate-arrays visited: 03.01.2026).

ДОДАТОК А

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Тіхонов С.В. Захищене кодування мережевих інтерфейсів у розподілених системах інтернету речей на основі багаторівневих формальних граматик. *Proc. of the VII International Scientific and Practical Conference «Science and Technology: Problems, Prospects and Innovations»* (Osaka, 13-15 April, 2023). С. 174-180. URL: https://sci-conf.com.ua/wp-content/uploads/2023/04/SCIENCE-AND-TECHNOLOGY-PROBLEMS-PROSPECTS-AND-INNOVATIONS-13-15.04.23.pdf.

2. Tikhonov Serhii. Hierarchical data coding approach based on adaptive formal grammar. *Proc. of the XIV International Scientific and Practical Conference "Science and Practice: Implementation to Modern Society"* (Manchester, 26-28 April, 2023), № 152. P. 559-563. URL: https://archive.interconf.center/plugins/generic/pdfJsViewer/pdf.js/web/viewer.html?file=https%3A%2F%2Farchive.interconf.center%2Findex.php%2Fconference-proceeding%2Fissue%2Fdownload%2F26-28.04.2023%2F162#%5B%7B%22num%22%3A1540%2C%22gen%22%3A0%7D%2C%7B%22.

3. Ситніков В.С., Тіхонов С.В. Принципи захисту інформації у комп'ютерних системах Інтернету речей на основі ієрархічного кодування у формальних граматиках. *Збірник тез 24-ї Міжнародної НПК «Сучасні інформаційні та електронні технології» CIET/MIET-2023* (Одеса, 29-31 травня 2023). С. 31-32. URL: https://old.tkea.com.ua/siet/archive/2023/31.pdf.

4. Тіхонов С.В. Формальна граматика машини Тьюринга для завадостійкого кодування фреймів Ethernet на основі трійкових лінійних сигналів. *Proc. of XII Internetional Scientific and Practical Conference «Innovations and Prospects in Modern Science»* (Stockholm, 20-22 November, 2023). С. 315-325. URL: https://sci-conf.com.ua/wp-content/uploads/2023/11/INNOVATIONS-AND-PROSPECTS-IN-MODERN-SCIENCE-20-22.11.23.pdf.

5. Тіхонов С. Питання кібербезпеки в базових технологіях інтернету речей. *Proc. of XII International Scientific and Practical Conference "Current Challenges of Science and Education"* (Berlin, 29-31 July, 2024). С. 165-171. URL: https://sci-conf.com.ua/wp-content/uploads/2024/07/CURRENT-CHALLENGES-OF-SCIENCE-AND-EDUCATION-29-31.07.24.pdf.

6. Тіхонов В., Тіхонов С., Яворська О. Дослідження комбінаторних потокових задач на планарному графі комп'ютерної мережі. *Proc. of XI International Scientific and Practical Conf. "Science in the Modern World: Innovations and Challenges"* (Toronto, 10-12 July 2025). С. 45-52. URL: https://sci-conf.com.ua/wp-content/uploads/2025/07/SCIENCE-IN-THE-MODERN-WORLD-INNOVATIONS-AND-CHALLENGES-12-14.06.2025.pdf.

7. Тіхонов С.В. Підвищення стійкості ad-hoc мережі IoT на основі швидкого алгоритму розподілу даних з використанням ШІ. *Proc. of V International Scientific and Practical Conference "International Experience in Scientific Rresearch"* (Chicago, 18-20 December 2025). P. 300-308. URL: https://sci-conf.com.ua/wp-content/uploads/2025/12/INTERNATIONAL-EXPERIENCE-IN-SCIENTIFIC-RESEARCH-18-20.12.25.pdf.

8. Tikhonov V., Taher A., Tikhonov S., Shulakova K., Hluschenko V., Chaika A. Turing Machine Development for High-secure Data Link Encoding in the Internet of Things Channel. *Hochschule Anhalt Edition "Applied Innovations in Information and Communication Technology" (ICAIIT)*. 2024. V. 12, № 1. P. 1-10. ISSN 2199-8876. DOI: http://dx.doi.org/10.25673/115635.

9. Tikhonov V., Siemens E., Vasiliu Y., Sitnikov V., Taher A., Tykhonova O., Shulakova K., Tikhonov S. Context-Defined Model of Open Systems Interaction for IoT Cybersecurity Issues Study. *Hochschule Anhalt Edition "Proceedings of the International Conference on Applied Innovations in IT" (ICAIIT).* 2024. V. 12, № 2, P. 35–44. ISSN 2199-8876. DOI: http://dx.doi.org/10.25673/118112.

10. Tikhonov V., Taher A., Shulakova K., Tikhonov S., Demchenko O. Turing Machine Development for High Protected Remote Control of the IoT Mobile Platform. *Springer Nature Link, Lecture Notes in Networks and Systems, ICAIIT- 2024*. 2025. V. 1338. P. 61–80. ISSN 2367-3370, 2367-3389 (electronic). DOI: https://doi.org/10.1007/978-3-031-89296-7_5.

11. Tikhonov V., Sitnikov V., Tikhonov S. An Improved AI-Driven Algorithm of Data Flow Optimization on the Normalized Bipolar Planar Free-Oriented Network Graph. *Hochschule Anhalt Edition "Proceedings of the International Conference on Applied Innovations in IT" (ICAIIT).* 2025. V. 13, № 5. P. 7–14. ISSN 2199-8876. DOI: http://dx.doi.org/10.25673/122800. URL: https://www.icaiit.org/paper.php?paper=13th_ICAIIT_5/1_2.

ДОДАТОК Б

АКТ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ ЗДОБУВАЧА НА ПІДПРИЄМСТВІ ТОВ «РАДУГА-N»

АКТ

впровадження на підприємстві ТОВ «Радуга-N» м. Одеса
результатів дисертаційної роботи Тіхонова С.В. на тему:
МЕТОД КРИПТО-СТІЙКОГО ДИНАМІЧНОГО КОДУВАННЯ
І РОЗПОДІЛУ ДАНИХ МОБІЛЬНОЇ AD-HOC МЕРЕЖІ
ІНТЕРНЕТУ РЕЧЕЙ

Даним актом засвідчуємо, що на підприємстві ТОВ «Радуга-N» м. Одеса у період з жовтня по грудень 2025 року, в рамках реалізації послуг «Розумний дім», запроваджено спосіб динамічної синхронізації поточного профільного ключа між контролером мобільного об'єкту і базовою станцією, який запропоновано в дисертаційній роботі Тіхонова С.В «Метод крипто-стійкого динамічного кодування і розподілу даних мобільної ad-hoc мережі інтернету речей».

Даний спосіб використовується для програмування радіо-пультів і систем управління автоматизованими пристроями Інтернету речей у приватних оселях та підприємствах (ролетами, гаражними воротами тощо) з метою підвищення їхньої стійкості до зламу.

Директор ТОВ «Радуга-N» _____ Лисейко О.І.

Головний інженер ТОВ «Радуга-N» _____ Огриза В.А.

17 грудня 2025 р.

ДОДАТОК В

АКТ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ ЗДОБУВАЧА В НАВЧАЛЬНИЙ ПРОЦЕС ДУІТЗ

«ЗАТВЕРДЖУЮ»
декан факультету Інформаційних технологій та кібербезпеки
Державного університету інтелектуальних технологій і зв'язку,
д.т.н., професор

_____ Василіу Є.В.
«_25__»_грудня_____ 2025 р.

АКТ
впровадження в навчальний процес
наукових праць аспіранта кафедри Комп'ютерних систем
Національного університету «Одеська Політехніка»
Тіхонова Сергія Вікторовича

Комісія у складі
голови комісії – д.т.н., проф. Корчинського В.В., зав. кафедри кібербезпеки та технічного захисту інформації
і членів комісії:
к.т.н., доц. Кільдишева Віталія Йосиповича, доцента каф. КБ та ТЗІ ДУІТЗ,
к.т.н., Рябухи О.М., ст. викл. каф. КБ та ТЗІ ДУІТЗ
у період з 23 по 25 грудня 2025 р. розглянула матеріали наукових праць аспіранта кафедри Комп'ютерних систем Національного університету «Одеська Політехніка» Тіхонова С.В., опубліковані у наукових фахових виданнях, що індексуються в базі даних Scopus, а також окремі результати його дисертаційної роботи «Метод крипто-стійкого динамічного кодування і розподілу даних мобільної ad-hoc мережі Інтернету речей», яка представлена на здобуття наукового ступеня доктора філософії за спеціальністю 123 – Комп'ютерна інженерія.

Комісія встановила:

1) На кафедрі КБ та ТЗІ при читанні лекцій по дисциплінам, що вивчаються за спеціальністю «Кібербезпека та захист інформації» галузі знань «Інформаційні технології, є доцільним використання окремих матеріалів С.В. Тіхонова, опублікованих у наукових фахових виданнях, що індексуються в базі даних Scopus, а також окремих результатів його дисертаційної роботи «Метод крипто-стійкого динамічного кодування і розподілу даних мобільної ad-hoc мережі Інтернету речей».

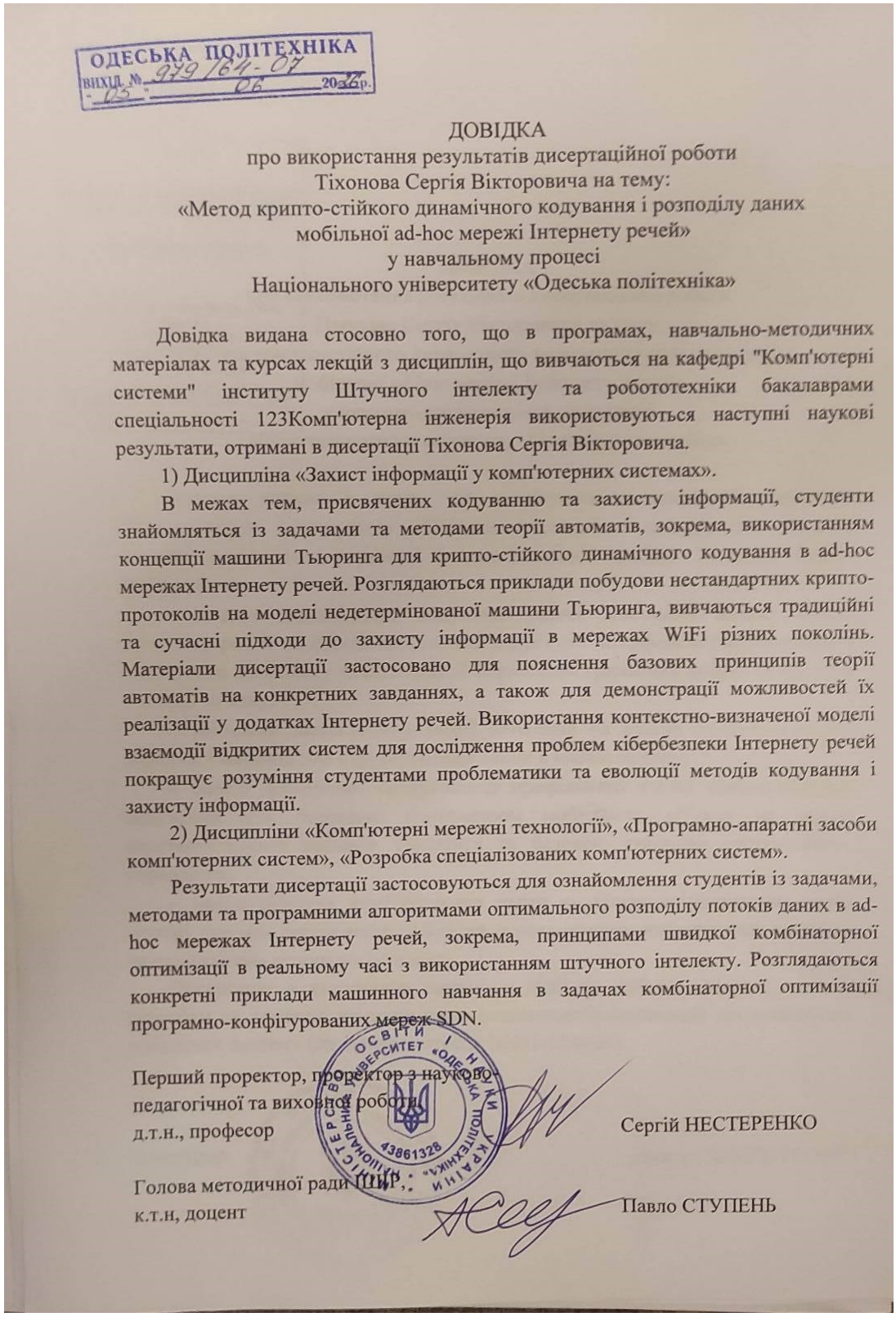
2) У дисциплінах «Теоретичні основи передавання та захисту інформації» та «Безпека і експлуатація мережевих і хмарних технологій» рекомендовано ввести розділи «Методи теорії автоматів і недетермінованої машини Тьюринга для крипто-стійкого кодування в ad-hoc мережах Інтернету речей» та «Контекстно-визначена модель взаємодії відкритих систем для дослідження проблем кібербезпеки Інтернету речей».

3) У дисципліну «Моніторинг та аудит інформаційно-комунікаційних систем» рекомендовано ввести розділ «Алгоритм комбінаторної оптимізації розподілу даних з використанням штучного інтелекту».

Голова комісії, д.т.н., проф. _____ Корчинський В.В
Члени комісії:
к.т.н., доцент _____ Кільдишев В.О.
к.т.н, ст. викл. _____ Рябуха О.М.

ДОДАТОК Г

ДОВІДКА ПРО ВИКОРИСТАННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ ЗДОБУВАЧА У НАВЧАЛЬНОМУ ПРОЦЕСІ НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ «ОДЕСЬКА ПОЛІТЕХНІКА»



ОДЕСЬКА ПОЛІТЕХНІКА
ВИХІД № 979/64-07
06 2026 р.

ДОВІДКА
про використання результатів дисертаційної роботи
Тіхонова Сергія Вікторовича на тему:
«Метод крипто-стійкого динамічного кодування і розподілу даних мобільної ad-hoc мережі Інтернету речей»
у навчальному процесі
Національного університету «Одеська політехніка»

Довідка видана стосовно того, що в програмах, навчально-методичних матеріалах та курсах лекцій з дисциплін, що вивчаються на кафедрі "Комп'ютерні системи" інституту Штучного інтелекту та робототехніки бакалаврами спеціальності 123Комп'ютерна інженерія використовуються наступні наукові результати, отримані в дисертації Тіхонова Сергія Вікторовича.

1) Дисципліна «Захист інформації у комп'ютерних системах».

В межах тем, присвячених кодуванню та захисту інформації, студенти знайомляться із задачами та методами теорії автоматів, зокрема, використанням концепції машини Тьюринга для крипто-стійкого динамічного кодування в ad-hoc мережах Інтернету речей. Розглядаються приклади побудови нестандартних крипто-протоколів на моделі недетермінованої машини Тьюринга, вивчаються традиційні та сучасні підходи до захисту інформації в мережах WiFi різних поколінь. Матеріали дисертації застосовано для пояснення базових принципів теорії автоматів на конкретних завданнях, а також для демонстрації можливостей їх реалізації у додатках Інтернету речей. Використання контекстно-визначеної моделі взаємодії відкритих систем для дослідження проблем кібербезпеки Інтернету речей покращує розуміння студентами проблематики та еволюції методів кодування і захисту інформації.

2) Дисципліни «Комп'ютерні мережні технології», «Програмно-апаратні засоби комп'ютерних систем», «Розробка спеціалізованих комп'ютерних систем».

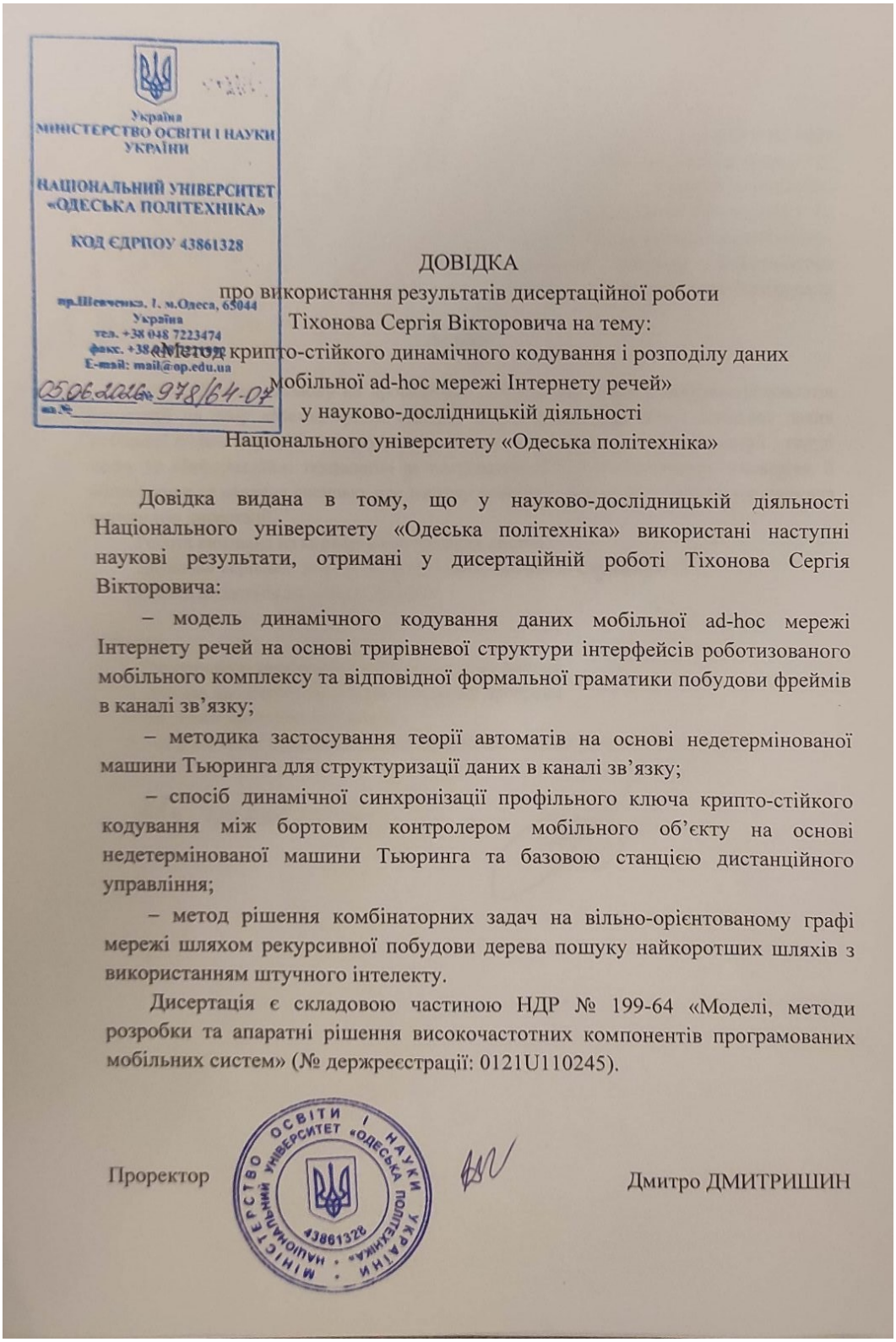
Результати дисертації застосовуються для ознайомлення студентів із задачами, методами та програмними алгоритмами оптимального розподілу потоків даних в ad-hoc мережах Інтернету речей, зокрема, принципами швидкої комбінаторної оптимізації в реальному часі з використанням штучного інтелекту. Розглядаються конкретні приклади машинного навчання в задачах комбінаторної оптимізації програмно-конфігурованих мереж SDN.

Перший проректор, проректор з науково-педагогічної та виховної роботи, д.т.н., професор — Сергій НЕСТЕРЕНКО

Голова методичної ради ПНІР, к.т.н, доцент — Павло СТУПЕНЬ

ДОДАТОК Д

ДОВІДКА ПРО ВИКОРИСТАННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ ЗДОБУВАЧА У НАУКОВО-ДОСЛІДНИЦЬКІЙ ДІЯЛЬНОСТІ НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ «ОДЕСЬКА ПОЛІТЕХНІКА»



Україна
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»

КОД ЄДРПОУ 43861328

пр.Шевченка, 1, м.Одеса, 65044
Україна
тел. +38 048 7223474
факс. +38 048 7223474
E-mail: mail@op.edu.ua

05.06.2026 № 978/64-07

ДОВІДКА
про використання результатів дисертаційної роботи
Тіхонова Сергія Вікторовича на тему:
«Метод крипто-стійкого динамічного кодування і розподілу даних мобільної ad-hoc мережі Інтернету речей»
у науково-дослідницькій діяльності
Національного університету «Одеська політехніка»

Довідка видана в тому, що у науково-дослідницькій діяльності Національного університету «Одеська політехніка» використані наступні наукові результати, отримані у дисертаційній роботі Тіхонова Сергія Вікторовича:

– модель динамічного кодування даних мобільної ad-hoc мережі Інтернету речей на основі трирівневої структури інтерфейсів роботизованого мобільного комплексу та відповідної формальної граматики побудови фреймів в каналі зв'язку;

– методика застосування теорії автоматів на основі недетермінованої машини Тьюринга для структуризації даних в каналі зв'язку;

– спосіб динамічної синхронізації профільного ключа крипто-стійкого кодування між бортовим контролером мобільного об'єкту на основі недетермінованої машини Тьюринга та базовою станцією дистанційного управління;

– метод рішення комбінаторних задач на вільно-орієнтованому графі мережі шляхом рекурсивної побудови дерева пошуку найкоротших шляхів з використанням штучного інтелекту.

Дисертація є складовою частиною НДР № 199-64 «Моделі, методи розробки та апаратні рішення високочастотних компонентів програмованих мобільних систем» (№ держреєстрації: 0121U110245).

Проректор Дмитро ДМИТРИШИН

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ · НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА» · 43861328

ДОДАТОК Е

МЕТОДИКА І РОЗРАХУНОК ЕФЕКТИВНОСТІ МЕТОДУ КРИПТО-СТІЙКОГО ДИНАМІЧНОГО КОДУВАННЯ І РОЗПОДІЛУ ДАНИХ МОБІЛЬНОЇ AD-HOC МЕРЕЖІ IoT

Викладається методика і розрахунки ефективності крипто-стійкого динамічного кодування і розподілу даних ad-hoc мережі IoT у 3-х аспектах.

I. Ефективність використання каналу зв'язку порівняно з технологіями WiFi різних поколінь від WiFi-1 (802.11) до WiFi-7 (802.11be).

II. Ефективність динамічного кодування даних порівняно з відомими методами захисту і шифрування інформації.

III. Ефективність алгоритму динамічного розподілу даних в ad-hoc мережі IoT порівняно з відомими методами оптимізації цифрових потоків в мережах.

I. Ефективність використання каналу зв'язку

Відомими показниками використання каналу зв'язку є енергетична, частотна та інформаційна ефективність. Перші два характеризують фізичний рівень (сигнально-кодові конструкції і модуляцію), що не є предметом розгляду у даній дисертації. Інформаційна ефективність має теоретичну оцінку

$$\eta = R/C, \text{ де} \qquad (1)$$

R – швидкість передачі корисної та службової інформації (біт/с); C – ємність (пропускна здатність) каналу. Для аналогового каналу з адитивним Гаусовим шумом, ємність C у формулі (1) визначається за формулою Шеннона-Гартлі

$$C = \Delta F \cdot \log_2(1+SNR), \text{ де} \qquad (2)$$

ΔF – смуга частот пропускання каналу (Гц); SNR=S/N (Signal to Noise Ratio) – співвідношення «сигнал/шум»; S – потужність корисного сигналу; N – потужність шуму [1].

Відзначимо наступне.

1) В мобільних мережах параметри SNR та ΔF не є стабільними і однаковими, тому оцінка інформаційної ефективності η по формулах (1) і (2), є окремою достатньо складною експериментально-дослідницькою задачею.

2) Параметр R у формулі (1) характеризує фізичний рівень каналу незалежно від змісту інформації що вкладається у фрейм. Натомість, стек протоколів TCP/IP передбачає багаторівневу інкапсуляцію протокольних одиниць даних (PDU), кожен з яких додає свою службову інформацію у відповідних PDU, як показано нижче на рисунку 1. Тому *фактична інформаційна ефективність* («кпд») каналу залежить також від частки корисної інформації фрейму в загальному цифровому потоці R.

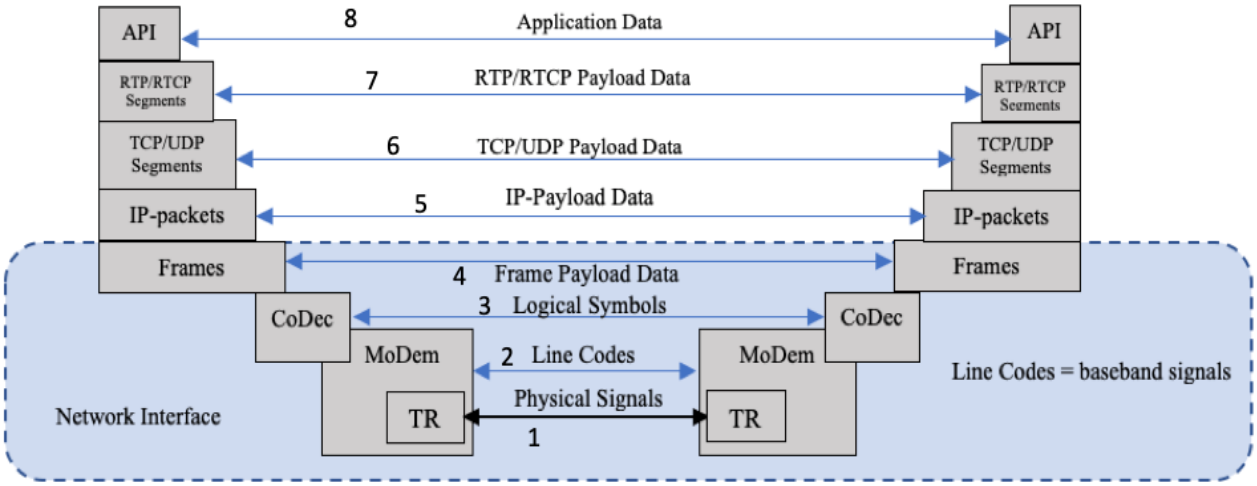


Рисунок 1 – Схема багаторівневої інкапсуляції протокольних одиниць даних (PDU) в стеку TCP/IP

Іншим важливим фактором, що впливає на ефективність використання каналу зв'язку WiFi, є обмеження на часовий інтервал утримання радіоканалу (слот Airtime порядку декількох мілісекунд) для передачі фрейму та очікування підтвердження (ACN) про його отримання. Цей фактор є суттєвим, якщо в зоні дії мобільного об'єкту працюють інші аналогічні об'єкти на одній частоті (тобто, спільний канал зв'язку розділяється між багатьма користувачами). Така ситуація є типовою для групи безпілотних мобільних об'єктів, або для локалізованого центру паралельного управління декількома окремими об'єктами різними пілотами.

Структуризація даних у фреймах проводової технології Ethernet та її аналогу (радіо-Ethernet) – технології WiFi сімейства стандартів IEEE 802.11 – мають спільні риси, натомість, відрізняються на канальному і фізичному рівнях.

Фрейм проводового Ethernet стандарту IEEE 802.3x (1997) показано на рисунку 2.

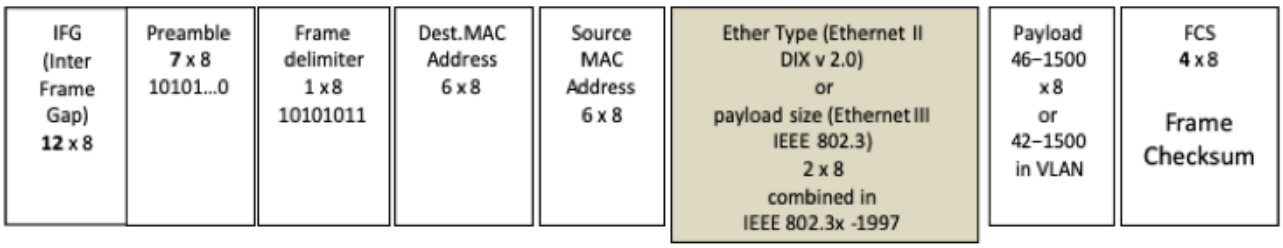


Рисунок 2 – Фрейм проводового Ethernet (стандарт IEEE 802.3x, 1997)

Стандарт проводового Ethernet IEEE 802.3x (Рис. 2) об’єднує два найбільш популярних стандарти: DIX v 2.0 (Ethernet-2) та IEEE 802.3 (Ethernet-3). Поле корисного навантаження “Payload” (PL) у фреймах проводового Ethernet обмежено розміром PL=(46÷1500) байт (окрім спеціальних великих фреймів типу “Jambo”). Якщо фактичний обсяг корисних даних менше 46 байт, поле PL доповнюється довільними бітами, напр., нульовими (т. зв. ‘padding”).

Особливості взаємодії сторін через проводові канали Ethernet на підрівні MAC (у т. ч., через канал спільного користування в режимі полу-дуплекс) передбачають, що передавач має робити обов’язкову паузу після кожного переданого фрейму (Inter Frame Gap, IFG) розміром не менше 12 умовних байт: $IFG \geq (12 \times 8) \cdot \Delta t$, де Δt – інтервал часу на передачу одного біту інформації. Зокрема, пауза IFG потрібна для «логічної синхронізації» потоку даних – відокремлення наступного фрейму від попереднього на приймальній стороні каналу. Фізичний розмір часового інтервалу IFG залежить від конкретної стандартизованої швидкості передачі біт (наприклад, 100 Мбіт/с для Fast Ethernet) [2].

Для підвищення інформаційної ефективності використання радіоканалу, у стандартах WiFi починаючи від покоління WiFi-4 (2009) до сучасного WiFi-7 (специфікації від 802.11n до 802.11ax/be), застосовують збільшений до 2346 байт розмір фрейму і розширеним до 2304 байт полем корисного навантаження. При цьому, хоча WiFi MAC дозволяє поле корисного навантаження до 2304 байт для інкапсуляції IP-пакету, на рівні IP зазвичай стоїть стандартний ліміт 1500 байтів.

Окрім того, покоління WiFi (починаючи від WiFi-4, 2009), передбачають можливість агрегації фреймів шляхом заповнення часового слоту Airtime двома чи більше послідовними фреймами. За рахунок цього, зменшується відносна частка службових пауз між фреймами у загальному часі використання радіоканалу, і збільшується його інформаційної ефективність. Натомість, зворотною стороною агрегації фреймів є погіршення динаміки взаємодії між об’єктами через збільшення періоду часу на оновлення даних телеметрії.

Специфікація WiFi IEEE 802.11n (2009) визначає два основні типи фреймів WiFi для агрегації:

1) MSDU (MAC Service Data Unit) – розширений і адаптований під радіоканал WiFi фрейм проводового стандарту Ethernet без шифрування;

2) MPDU (MAC Protocol Data Unit) – розширений для цілей крипто-захисту фрейм типу MSDU, який містить додаткові 46–50 байтів на шифрування даних (в залежності від конкретного стандарту захисту WiFi).

Структура WiFi-фрейму типу MSDU на першому підрівні MAC канального рівня L2 OSI будується на основі стандартного IP-пакету протоколу версії IPv4, який формується на верхніх рівнях L3÷L7 OSI, і передається на другий підрівень LLC канального рівня L2 OSI, як показано на рисунку 3 [3]. Драйвер підрівня LLC перетворює поле корисного навантаження IP-пакету розміром (46÷1500) байт у розширене поле корисного навантаження типу "MSDU payload" розміром (0÷2304) байт, в якому вже немає обмеження на мінімальний розмір 46 байт для поля PL проводового стандарту Ethernet.

Далі, поле "MSDU payload" доповнюється 8 байтами заголовку LLC/SNAP (Sub-Network Access Protocol) для сумісності із застарілими версіями протоколу IPv4 (такими як IPX, AppleTalk тощо), і перетворюється у так званий "Frame Body" (тіло фрейму розміром 0÷2012 байт) підрівня MAC.

Нарешті, драйвер MAC-рівня додає ліворуч до тіла фрейму заголовок "MAC Header" і праворуч «хвостовик» (Trailer) – контрольна сума фрейму FCS (Frame Check Sum). Заголовок фрейму має обов'язкові три перші поля (Frame Control, Duration/ID, Address 1) розміром 2+2+6=10 байт, а також чотири необов'язкові поля загальним розміром від 18 до 18+(2+6+2+4)= 32 байти. Поле контрольної суми зберігає розмір 4 байти (як у проводовому фреймі Ethernet).

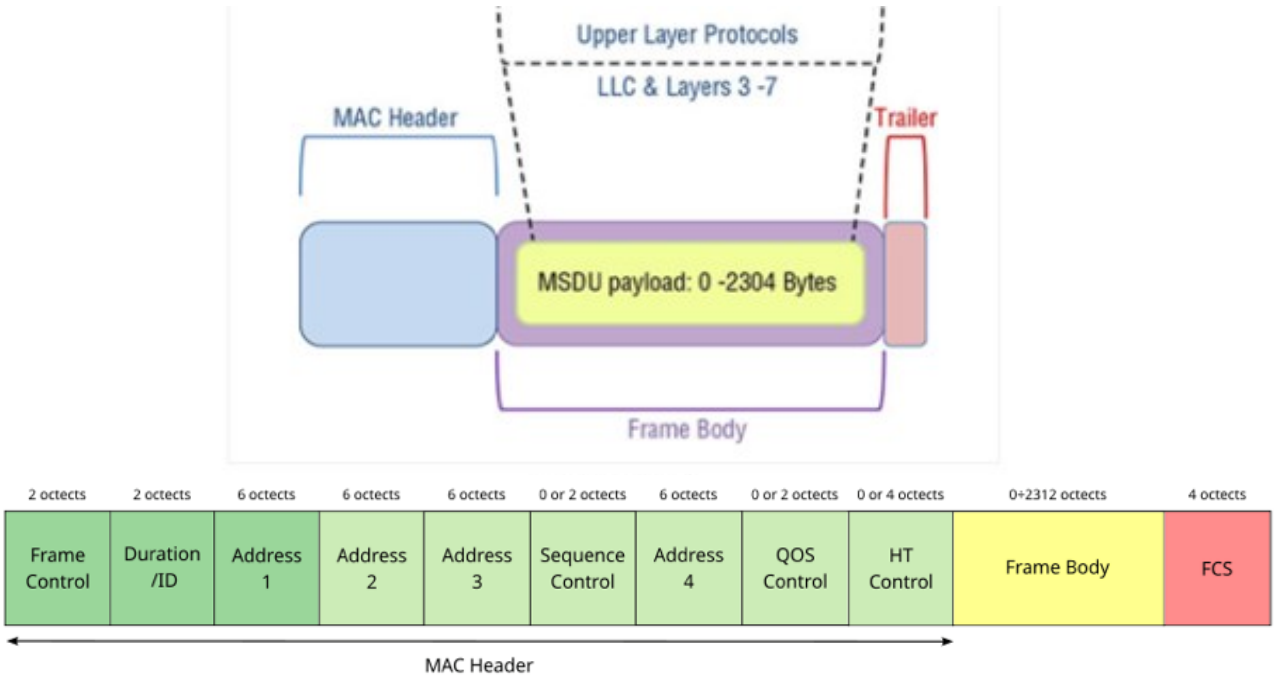


Рисунок 3 – Структура WiFi фрейму типу MSDU

Поле “Frame Control” (Рис. 3) містить бітові «прапорці» для визначення типу фрейму. Стандарт 802.11-2016 передбачає наступні типи фреймів [3].

Beacon frame (фрейм-маяк) від точки доступу WiFi що повідомляє клієнтів про себе і свій частотний канал. Дозволяє клієнту зі спрямованою антеною направити її на «маяк» для найкращого зв’язку.

Association frame (фрейм-hello) від клієнта що повідомляє точку доступу WiFi про свою готовність і намір надіслати їй фрейм даних для прийому.

Authentication frames “AUTH_Request”, “AUTH_Response” від клієнтів і точки доступу WiFi, які використовуються для автентифікації клієнтів (підтвердження того, що клієнт з даною MAC-адресою є насправді тим, за кого себе видає).

- Data frame (фрейми з корисними даними).

- Acknowledgement (ACK) frame: короткий фрейм-підтвердження успішного прийому від передавача.

- BlockACK frame: підтвердження прийому окремих блоків фрейму від передавача (дієвий механізм оцінки інтенсивності і планування трафіка в радіоканалі).

Двобайтове поле “Duration/ID” визначає тривалість часу (у мікросекундах), на який утримується радіоканал для передачі фрейму адресату, очікування від нього відповіді про успішне отримання фрейму (ACK), а також для додаткового інтервалу SIFS (Short Interframe Space), який необхідний адресату на обробку отриманого фрейму перед відправленням підтвердження про його успішний прийом [4].

Нестандартні Raw Socket фрейми Ethernet/WiFi.

Дієвим інструментом для підвищення інформаційної ефективності використання каналів зв’язку проводового Ethernet та його бездротового аналогу WiFi є застосування так званих «сирих з’єднань» (Raw Socket connection) з нестандартними протоколами і структурами фреймів на канальному рівні (Data Link Layer, DLL), які передбачені в Unix-подібних мережевих операційних системах Unix, FreeBSD, Linux, MAC [5]. Технологія Raw Socket може бути реалізована програмно на більшості діючих мережевих інтерфейсах Ethernet та радіо-інтерфейсах WiFi. Такий варіант передбачається, у тому числі, для запропонованого в дисертаційній роботі методу крипто-стійкого динамічного кодування даних в мобільній ad-hoc мережі Інтернету речей.

Оцінимо вплив службової інформації WiFi-фрейму типу MSDU на ефективність радіоканалу. Визначимо частку корисної інформації фрейму:

$$\mu = Q_U / Q_F \text{, де} \quad (3)$$

Q_U (біт) – обсяг корисної інформації у складі фрейму, Q_F (біт) – загальний обсяг усіх полів фрейму на підрівні MAC канального рівня.

З урахуванням цього, фактична (з точки зору частки корисної інформації μ) інформаційна ефективність каналу (позначимо її η') матиме вигляд:

$$\eta' = \mu \cdot \eta = (\mu \cdot R)/C. \quad (4)$$

Нехай:

μ – частка корисної інформації деякого нестандартного фрейму WiFi, який передається зі швидкістю R по радіоканалу ємності C;

μ_0 – частка корисної інформації деякого стандартного фрейму WiFi, який обрано для порівняння в однакових умовах (швидкість передачі R по радіоканалу ємності C);

η'_0 – фактична (з точки зору частки корисної інформації μ_0) інформаційна ефективність каналу з обраним для порівняння стандартним фреймом WiFi.

Визначимо відносну інформаційну ефективність (позначимо її β) для порівняння двох способів кодування фреймів (з параметрами μ та μ_0) при використанні каналу однакової ємності C на швидкістю передачі R:

$$\beta = \eta'/\eta'_0 = [(\mu \cdot R)/C] / [(\mu_0 \cdot R)/C] = \mu / \mu_0 \text{, або}$$

$$\beta = \mu / \mu_0 . \quad (5)$$

Як видно з формули (4), відносна інформаційна ефективність β залежить тільки від структури фреймів що порівнюються між собою, і не залежить від конкретної швидкості R передавання даних по каналу зв’язку та його фактичної ємності C в окремі моменти часу.

Для зручності розрахунку відносної інформаційної ефективності β по формулі (4) для запропонованого в дисертаційній роботі методу крипто-стійкого динамічного кодування даних в мобільній ad-hoc мережі Інтернету речей, зведемо в таблицю 1 поля заголовку фреймів для різних поколінь у сімействі стандартів WiFi, починаючи від WiFi-1 (802.11-1997) до перспективного (очікуваного) стандарту 802.11bn покоління WiFi-8.

Верхня частина таблиці описує поля фреймі типу MSDU (без шифрування); останні чотири строчки таблиці 1 описують поля заголовку фреймів з шифруванням даних.

Таблиця 1 – Поля заголовку фреймів WiFi сімейства IEEE 802.11

Структура неагрегованого фрейму WiFi у стандартах сімейства IEEE 802.11+
MSDU (MAC service data unit) - **нешифровані фрейми**. MDPU (MAC Protocol Data Unit) - шифровані фрейми

Поле	Size (bytes)	Generation	Year	Стандарт	Призначення поля
		WiFi-1	1997	802.11	Застарілий початковий стандарт WiFi (зараз не підтримується)
FrameControl	2	WiFi-2	1999	802.11b WPA2 AES	Тип фрейму (data/control); версія протоколу; flag "To DS/From DS"
Duration / ID	2	WiFi-2	1999	802.11a WPA2 AES	Заявлена трив. сесії (мкс) NAV (Network Allocation Vector -local timer
Address 1	6	WiFi-3	2003	802.11g **WPA2**	**RA (Receiver Address)** - мак-адреса безпосереднього отримувача. WPA2: AES-128-bit шифр + PSK автентифікація + CCMP крипто-протокол
Address 2	6	WiFi-4	2009	802.11n **WPA2**	**TA (Transmitter Address)** - мак-адреса того, хто зараз передає. WPA2: AES-128-bit шифр + PSK автентифікація + CCMP крипто-протокол
Address 3	6	-"-	-"-	-"-	**DA/SA** - адреса кінцевого пункту призначення або джерела.
SequenceControl	2	-"-	-"-	-"-	Номер пакета та номер фрагмента (щоб відрізнити дублікати)
QoS Control	2	-"-	2005	802.11e	VLAN traffic priority: voice/video/data - сучасн. станд. обов. майже всюди
FCS	4	-"-	-"-	-"-	FCS- Frame Check Sequence (контрольна сума фрейму (CRC)
Address 4	6	WiFi-5	2013	802.11ac	Only in WDS (Wireless Distribution System: AP-to-AP)/Mesh: any-to-any
HT Control	4	WiFi-6	2021	802.11ax **WPA3**	High Throughput., OFDM, ≤9.6 Gbps. (802.11n/ac/ax - 4/4/6 байт) WPA3: AES-128/256 шифр +SAE (Dragonfly) автентиф.+CCMP/GCMP крипто-протокол.
EHT	6	WiFi-7	2024	802.11be **WPA3**	Extremely High Throughput. WPA3: AES-128/256 шифр+SAE (Dragonfly) автентиф.+CCMP/GCMP крипто-протокл.
expected		WiFi-8	Prospect	802.11bn	Перспективний стандарт
WPA2 or **WPA3** Personal	+16	WiFi-3/5			Додаткові службові поля для шифрування постандарту безпеки WPA2
AES CCMP Header before data	(+8)				CCMP заголовок на початку фрейму (Counter Mode CBC-MAC Protocol): first **2** bytes of **PN** (packet number); Reserved (**1** byte); key ID (7-bit) + Ext IV (1-bit flag of extended Init. Vector IV); last 4 bytes of PN (packet number).
MIC after data	(+8)				MIC (Message Integrity Code) - цифровий підпис у кінці фрейму
WPA3 Enterprise	(+8)				Додаткове розширення поля цифрового підпису MIC з 8 до 16 байт

Інформаційна ефективність методу крипто-стійкого динамічного кодування даних порівняно з технологією WiFi в незахищених режимах роботи

Показник відносної інформаційної ефективності $\beta=\mu/\mu_0$ у формулі (4) суттєво залежить від частки корисної інформації Q_U в загальній структурі фрейму розміром Q_F: $\mu=Q_U / Q_F$. Представимо загальний розмір фрейму Q_F як суму $Q_F=Q_U+OH$, де OH – надлишкова інформація (Overhead), яка складається з окремих службових полів у заголовках різних рівнів інкапсуляції даних (див. Рис. 1). При заданому однаковому розмірі Q_U для μ та μ_0, маємо:

$$\beta = \mu/\mu_0 = (Q_U / Q_F) / (Q_U / Q_{F0}) = Q_{F0} / Q_F, \text{ або}$$

$$\beta = (Q_U + OH_0) / (Q_U + OH). \quad (6)$$

При малих Q_U, параметр $\mu=Q_U/Q_F$ та ефективність каналу $\eta'=\mu\cdot\eta$ у формулах (3)-(4), можуть бути критично малими. Натомість, саме такий випадок є характерним в задачах дистанційного управління мобільними об'єктами в режимі реального часу, коли затримки в передачі даних є критичними, а частота оновлення даних телеметрії і команд управління має бути якомога вище.

З іншого боку, надлишковість фреймів WiFi, як параметр OH_0 у формулі (6), має тенденцію збільшуватися у кожному наступному поколінні стандартів WiFi (див. Таб. 1). З урахуванням сказаного вище, оцінимо інформаційну ефективність β на типовому прикладі розмітки потоку даних в каналі зв'язку командами машини Тьюринга (профіль «оперативна 3-канальна взаємодія в реальному часі», підрозділ 2.3 дисертації, с. 109). Даний профіль описує послідовність окремих фреймів **TMF** в межах однієї сесії, де кожен фрейм містить тіло **B** з трьох сегментів даних реального часу (RTD) для управління трьома параметрами мобільного об'єкту (наприклад, висота, швидкість та азімут дрона). Профіль передбачає відомі взаємодіючим сторонам адреси відправника та одержувача, незмінні в межах однієї сесії. Розмір сегменту RTD – від 0.5 байт і вище. Фрейм також має контрольну суму FC (7 біт). Криптозахист відсутній.

$$\begin{cases} \{...\} := \{\mathbf{TMF}\ \mathbf{TMF}\ ...\ \}; \\ \quad \mathbf{TMF} = [\mathbf{B}\ (\text{RTD RTD RTD})\ \mathbf{FC}] \\ \text{RTD} = \text{“bbbb...b”} \end{cases} \quad (7)$$

Розрахуємо загальну кількість службової інформації в одному фреймі **TMF**. Для цього, представимо теги розмітки потоку двійковими кодами таблиці 2 (див. рисунок 13 дисертації) з урахуванням синтаксичного знаку пробіл «_».

Таблиця 2 – Система команд прикладної машини Тюринга

№	Код	Мнемо	Семантика	Semantics
1	0	.	Точка	Dot
2	1	,	Кома	Comma
3	00	;	Крапка з комою	Semicolon
4	01	:	Двокрапка	Colon
5	10	(	Ліва дужка	Left Parenthesis
6	11	)	Права дужка	Right Parenthesis
7	000	H	Заголовок	Header
8	001	B	Тіло	Body
9	010	FC	Контрольна сума	Frame Check sequence
10	011	[	Ліва скобка	Left Bracket
11	100	]	Права скобка	Right Bracket
12	101	{	Ліва фігурна скобка	Left Brace
13	110	}	Права фігурна скобка	Right Brace
14	111	**$**	Розширення команд ≥ 4 біт)	Extension

TMF = [B (RTD RTD RTD) FC ктр.сума]

TMF = _011_001_10_bb...b_ bb...b_ bb...b_11_010_0000000_100_

Фрейм TMF містить 32 службових трійкових символи. Кількість службової інформації OH складає $32\times\log_2 3 \approx 32\times 1.6 \approx 51.2$ біт, або OH ≈ 6.4 байти.

Оберемо для порівняння 3 варіанти інкапсуляції даних реального часу (RTD) у фрейми WiFi типу MSDU згідно стеку TCP/IP (Рис. 1).

Спрощений варіант. RTD вкладено безпосередньо в поле “Frame Body” (розміром 0÷2312 байт) фрейму WiFi типу MSDU (Рис. 3). Службова інформація фреймів WiFi різних поколінь представлена в таблиці 1. Оберемо для порівняння стандарт 802.11n покоління WiFi-4 (2009). Даний тип використовує 3 службові поля попередніх поколінь, та додає нових поля. Загалом: OH_0 =24 байти.

Проміжний варіант. RTD вкладено в сегмент UDP (Рис. 1). Заголовок UDP (8 байт) збільшує надлишковість до 24+8=32 байти. Загалом: OH_0 =32 байти.

Стандартний варіант. RTD вкладено в поле “Payload” протоколу RTP (Real Time Protocol), який завжди працює разом з RTCP (Real Time Control Protocol). Мінімальний заголовок RTP (12 байт) плюс заголовок RTCP (4 бати) збільшують надлишковість фрейму WiFi до 32+(12+4)=48 байт. Загалом: OH_0 = 48 байт.

Опишемо відносну інформаційну ефективність β сіменством функцій $\beta(Q_U)$ від аргументу Q_U з параметром OH_0, графіки яких представлено на рисунку 4.

$$\beta(Q_U) = (Q_U + OH_0) / (Q_U + 6.4); \quad OH_0 \in [24, 32, 48]. \qquad (8)$$

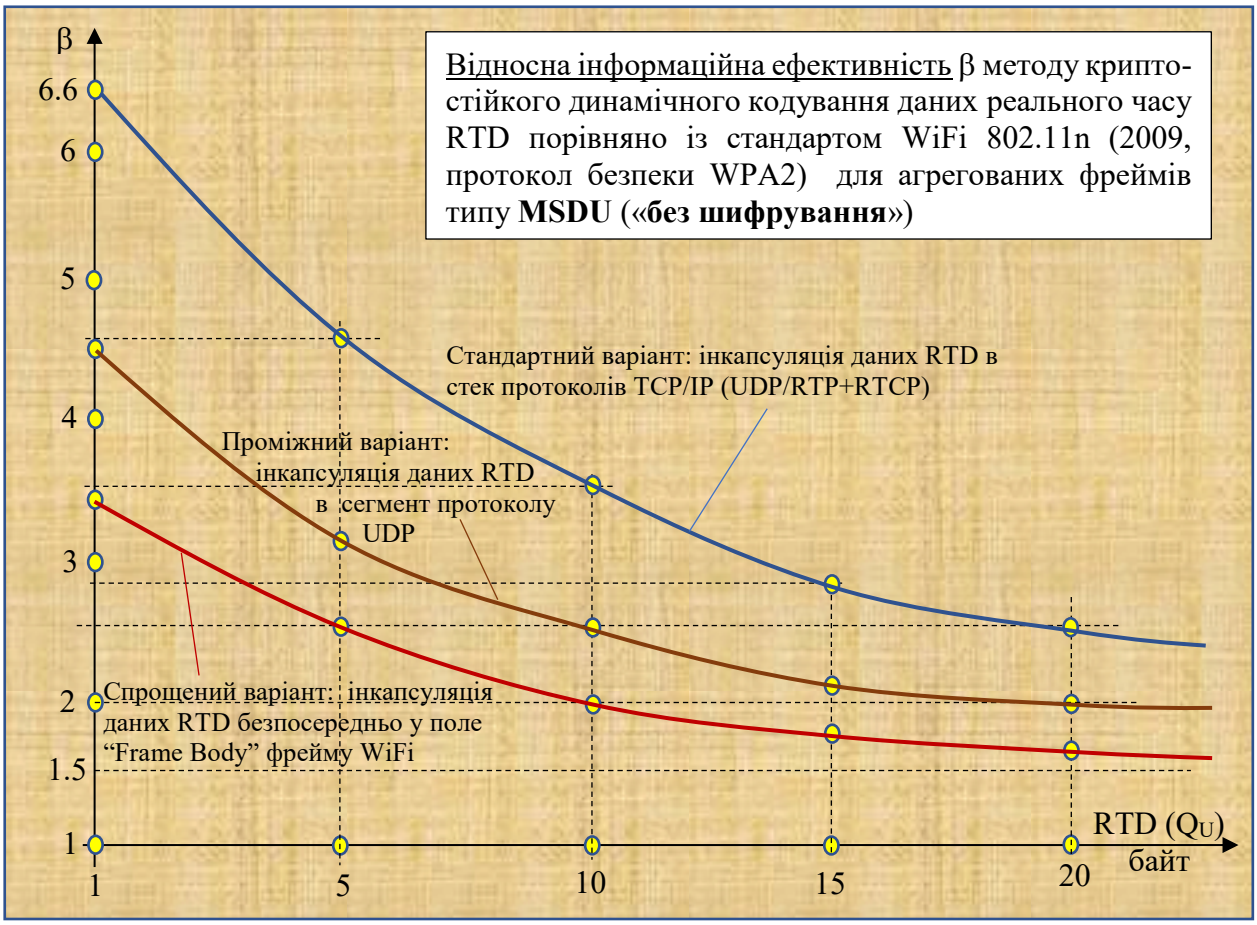


Рисунок 4 – Параметричне сімейство функції інформаційної ефективності β

Проведений вище порівняльний аналіз показників інформаційної ефективності засвідчує наступне.

1. В режимі дистанційного управління безпілотним об’єктом мобільної ad-hoc мережі в реальному часі (наприклад, FPV-дроном по радіоканалу) за умови відсутності або зневажливо малої ймовірності кіберзагроз, трафік телеметрії може передаватися відкрито без шифрування з метою мінімізації надлишковості мережевих протоколів та підвищення інформаційної ефективності каналу зв’язку.

2. Для такого класу завдань, запропонований в дисертації метод крипто-стійкого динамічного кодування даних, може використовуватися у *спрощеному режимі без крипто-захисту*, а саме – з фіксованим набором із 14 команд машини Тьюринга (наприклад, базовим набором команд, таблиця 2 додатку, рисунок 13 дисертації), або з будь яким іншим постійним («перемішаним») набором базових команд (незмінним ключом шифрування CEK). Аналогічно, обрана для порівняльного аналізу технологія WiFi 802.11n використовується у цьому випадку в режимі агрегації WiFi-фреймів типу MSDU (без шифрування) для мінімізації мережевих протоколів та підвищення інформаційної ефективності каналу (див. таб. 2 додатку).

3. Запропонований в дисертаційній роботі метод крипто-стійкого динамічного кодування, дозволяє підвищити відносну інформаційну ефективність каналу зв’язку β в діапазоні від **1.5** до **6.6** разів порівняно з популярною сьогодні технологією WiFi 802.11n, – в залежності від розміру блоків корисних даних реального часу RTD, а також від типу інкапсуляції даних у фрейми WiFi (стандартний, проміжний та спрощений варіанти), – як показано на рисунку 4 додатку.

ІІ. Ефективність крипто-стійкого динамічного кодування даних

Оцінка ефективності захисту даних, зазвичай, базується на трьох складових: стійкість до зламу, ресурсовитрати, мережева надлишковість [6].

1. Стійкість до зламу. Оцінюється по трьох показниках: складність повного перебору варіантів (Brute-force); лавинний ефект (Avalanche effect); інформаційна ентропія. *Складність перебору* має бути якомога більшою. Визначається, як правило, розміром ключа k (у бітах) та оцінюється як 2^k. Наприклад, сучасний стандарт AES має довжину ключа 128 або 256 біт.

Лавинний ефект є бажаним зокрема у блочних шифрах, і полягає у тому, що зміна одного довільного біта ключа або вхідних даних має призводити до зміни близько 50% бітів вихідних даних. Інакше, вихідний потік може бути корельований із вхідним потоком, що спрощує прогнозування вхідних даних і злам шифру (атаки на зіткнення, на розширення довжини, атаки на прообраз).

Інформаційна ентропія оцінює актуальну кількість інформації $H(X_N)$ у блоці цифрових даних X_N з N потенційно можливими комбінаціями, кожна з яких має ймовірність появи $p(x_n)$. Чим ближче розподіл $p(x_n)$ до «білого шуму» (рівномірний розподіл), тим більша ентропія блоку даних X_N, і тим важче провести статистичний аналіз блоку даних з метою зламу. Формула ентропії Шеннона для блоку даних: $H(X_N)= -\sum_n p(x_n)\cdot\log_2 p(x_n)$, $n\in[1,N]$ біт.

Наприклад, підкидання монети 2 рази. Блок даних результату має два символи, кожен з яких може бути «0» чи «1» – всього чотири потенційно можливі комбінації: 00, 01, 10, 11. Якщо всі комбінації рівно-можливі, тобто $p(x_n)\equiv 0.25$, то $H(S_2)= -4\cdot[0.25\cdot\log_2(0.25)] = -4\cdot[0.25\cdot(-2)]= 2$ біти. У будь якому випадку нерівномірного розподілу ймовірностей $p(x_n)$, ентропія H буде меншою.

2. Ресурсовитрати. Оцінюється по трьох основних показниках:

– обчислювальна складність (кількість тактів процесора на шифрування 1 байта); відомо, що шифр AES-128/256 є швидким в апаратній реалізації (AES-NI), а деякі «легковажні» шифри (SPECK, SIMON) є ще швидшими на МК.

– Енергоспоживання (Дж/біт).

– Затримка часу (T_{crypto}, Latency) що додається до циклу передачі даних.

3. Мережева надлишковість (Encryption Overhead). Є зворотною стороною захисту і шифрування даних в каналах зв'язку. Вона пов'язана з додатковими полями фрейму, які негативно впливають на інформаційну ефективність каналу зв'язку (див. чотири останні строчки таблиці 1).

Далі ці три поняття розглядаються детальніше.

Стійкість до зламу методу динамічного крипто-кодування.

Розглянемо, що означає «зламати деякий текст», зашифрований відомим хакеру алгоритмом, перебираючі можливі значення ключа, і якого розміру та змісту має бути цей текст, щоб впевненно його «зламати»? У криптоаналізі це поняття пов'язано з терміном «Відстань єдиності» (Unicity Distance).

«Зламати текст T» – це знайти такий ключ K, який відображує T у текст T` що має певний «зміст» для обраної мови або протоколу. Для AES-128 кількість можливих ключів становить 2^{128} (приблизно 3.4×10^{38}). Для його гарантованого зламу (повним перебором) на класичних комп'ютерах потрібно більше трильона років. Витратною є «*вартість верифікації*»: треба не тільки перебрати 2^{128} ключів, але й виконати 10 раундів перетворень алгоритму AES на кожній ітерації та провести статистичний тест результатів T` на кожний пробний ключ [7].

Стосовно характеру і розміру тексту для зламу. Клод Шеннон ввів формулу для розрахунку розміру L тексту, необхідного для того, щоб кількість помилкових ключів наблизилась до нуля: L= k/R, де k – довжина ключа в бітах; R – надлишковість мови (Redundancy) – різниця між максимально можливою ентропією мови та її реальною ентропією. Для англійської мови R≈3.2 біта на символ; L=128/3.2=40 символів. Отже, при зламі шифрованого англійського тексту 40÷50 символів, знайдений ключ є правильним з ймовірністю ≈100%.

Яка статистика (відстань єдиності) потрібна для зламу цифрового каналу?

Сценарій А. "Known Plaintext Attack" (KPA).

Якщо відомо, що будь-який пакет Wi-Fi починається із заголовка, де перші байти – це Frame Control (наприклад, 0x08 0x01), то достатньо 2-4 байти відомого тексту, щоб відсіяти 99.999...% хибних ключів. Далі, якщо після дешифрування перші два байти збіглися з очікуваним заголовком протоколу, можна вважати успішний злам ключа. Щоб впевнено «зламати» шифр AES-128, хакеру достатньо мати текст мережевого трафіку 25÷30 байтів (з передбачуваним типом заголовку). Якщо ж передається чисто випадковий цифровий потік, то його в принципі неможливо зламати через нульову надлишковість R.

Сценарій Б. "Ciphertext Only Attack" (COA). Невідомо, що всередині. Використовуються статистичні методи: а) частотний аналіз (напр., розподіл ймовірності літер різних мов); б) індекс збігів (coincidence index) – впорядкованість тексту (низька для шуму він, висока для реального тексту).

Особливість запропонованого в роботі методу крипто-кодування полягає в тому, що поточний секретний ключ CEK (Current Encryption Key) змінюється через малі інтервали часу, починаючи від наперед узгодженого номера ключа $n_0 \in [1, N]$ в масиві з N ключей місії $P_M(N)$ (наприклад, автоматично по таймерах). Масив $P_M(N)$ випадково генерується і записується в пам'ять кожної пари взаємодіючих сторін на стартовому етапі місії. Поточний ключ CEK змінює бінарні коди тегів розмітки потоку, і заплутує структуру фреймів для зламу.

Щоб унеможливити пряме перехоплення окремих блоків корисних та службових даних (які мають розмір від 4 біт і більше) на фізичному рівні каналу з трійковим лінійним кодуванням без знання поточного секретного ключа CEK, використовується «маскування» даних тегом «крапка» (код «_0_» в таблиці кодів машини Тьюринга, див. рис. 13 дисертації). Тег «крапка» фрагментує блок даних так, щоб з обох сторін були частини не менше 2 біт. В процесі крипто-кодування, цей тег приймає випадкові значення (аналогічно усім іншим тегам) відповідно до поточного ключа CEK. На приймальній стороні, перш за все, видаляють усі теги «крапка» за допомогою ключа CEK, а потім вже аналізують цифровий потік і виділяють в ньому структурні елементи, зокрема, фрейми.

Окрім того, може здійснюватися пост-шифрування корисних та службових даних (на додаток і незалежно від інших традиційних методів шифрування) тимчасовим секретним ключем TEK (Temporary Encryption Key), що передається відправником по каналу зв'язку «умовно-відкритим» випадковим порядковим номером "n" ключа в масиві ключей місії $P_M(N)$. Цей масив є унікальним для кожного окремого об'єкту мобільної ad-hoc мережі. Ключ TEK може змінюватися на кожному черговому фреймі що передається по каналу.

За цих обставин, актуальність зламу каналу зв'язку з метою перехоплення управління конкретним мобільним об'єктом має сенс лише в межах часу дії поточного ключа CEK і тимчасового ключа TEK (якщо він застосовується). Тому для оцінки захисту методу крипто-кодування і пост-шифрування важливим фактором є швидкодія сучасних процесорних пристроїв РЕБ.

Мікропроцесори 8080/8051 мають тактову (кварцеву) частоту 2÷12 МГц. Так, Intel 8051 (12 МГц) має машинний цикл 12 тактів на одну «машинну інструкцію. Сучасні конвеєрні МП (Core i9, Ryzen 9, Apple M3) працюють на тактовій частоті f_{clc}=3÷5 ГГц, а їхній «машинний цикл» (етап виконання частини інструкції) зазвичай дорівнює одному такту ($T_{clc}=1/f_{clc}$). За рахунок паралельної (конвеєрної) обробки частин інструкції, уся інструкція може виконуватися за один машинний такт, тобто показник CPI (Cycles Per Instruction) становить 1.

Найсучасніші процесори можуть мати CPI менше одиниці (тобто, за один машинний такт виконувати декілька інструкцій). На тактовій частоті f_{clc} =5 ГГц один такт становить $T_{clc}= 1/(5\cdot10^9)=0.2\cdot10^{-9}$ с. Припустимо, що статистичний аналіз перехопленого зловмисником фрейму на кожне пробне значення ключа потребує 100 машинних інструкцій (насправді значно більше). Кнвеєрний процесор що виконує інструкції паралельно з один машинний такт, витрачає на обробку одного пробного ключа час $\Delta t = 100\cdot T_{clc}= 100\cdot 0.2\cdot10^{-9} = 20\cdot10^{-9}$ с.

Множина P(N) ключів у методі крипто-стійкого кодування визначається кількістю перестановок з 14 команд машини Тьюринга (див. с. 103 дисертації): P(N) = P(14) = 14! = 87’178’291’200. Кожна перестановка з 14 команд є ключом крипто-кодування структури фрейму та шифрування даних.

Перед стартом кожної місії, повний або частковий набір ключів $P_M(N)\subseteq P(N)$ випадково перемішуються і записуються в пам’ять кожної пари взаємодіючих мобільних об’єктів та базової станції. Розмір частки $P_M(N)$ впливає на тривалість місії з неповторними ключами. Для масиву ключів місії $P_M(N)$ = 0.01% P(N), за умови, що ключ динамічно змінюється через кожні 0.1 секунди і ніколи не повторюється протягом однієї місії, максимальна тривалість місії становить $T_M = P(N)\cdot0.01\%\cdot0.1) \approx$ 871’783 секунд ≈ 242 години ≈ 10 діб.

Незалежно від частки ключа в місії $P_M(N)$, для зламу каналу, в найгіршому випадку, треба зробити повний перебір множини ключів P(N), і для кожного з них провести статистичний аналіз знятого сегменту даних. Це потребує час $T_{kf} = \Delta t \cdot P(N) = 20\cdot10^{-9} \cdot 87’178’291’200 \approx$1740 с ≈ 30 хвилин. В середньому випадку, цей час, у першому наближенні, можна прийняти вдічі менше (15 хвилин).

Якщо йдеться про канал обміну телеметрії рухомого мобільного об’єкту, то злам каналу із затримкою 15 хвилин практично унеможливлює перехоплення контролю над об’єктом, оскільки ключі на повторюються, а їхній загальний обсяг в межах місії тривалістю до 10 діб вичерпується лише на 0.01%.

Витрати ресурсів на динамічне крипто-кодування даних.

1) Флеш-пам’яті для ключів місії. Визначається часткою $P_M(N)\subseteq P(N)$ від множини ключів P(N)=14! = 87’178’291’200, та кортежом для одного ключа (списку з 14 перемішаних чисел від 1 до 14). Для представлення чисел від 1 до 14 достатньо 4 біти на кожне з них. Отже, кортеж з 14 чисел потребує 56 біт (7 байт) пам’яті. Масив $P_M(N) = 0.01\% \cdot P(N) = 0.01\% \cdot 87’178’291’200 \approx$ 8’717’830. Отже, потрібна пам’ять для $P_M(N)$ складає 7(байт)×8’717’830≈61Мбайт. Стандартний дешевий чіп пам’яті 64Мб цілком задовольняє цю вимогу.

2) Затримка часу на крипто-кодування/декодування.

Запропонований в дисертації метод динамічного кодування даних на основі машини Тьюринга (підрозділ 2.3 дисертації) реалізує потокове шифрування (крипто-кодування) по окремих символах (див. розд. 2.2).

Вхідний потік даних x(t) – це послідовність байт, кожен з яких містить два символи по 4 біти, які мають десяткові значення 0, 1, 2, ..., 15). Чотирнадцять символів {1, 2, ..., 14} означають коди команд машини Тьюринга (теги розмітки цифрового потоку даних в каналі зв'язку); символ «0» означає один біт даних «0»; символ «15» означає біт даних «1».

Процес лінійного кодування без шиффрування. Вхідні символи x(t), які мають значення від 1 до 14 (теги розмітки потоку), перетворюються у вихідні трійкові коди машини Тьюринга y(t) відповідно до системи команд (див. рис. 2.6 дисертації), а символи «0» та «15» – у біти даних «0» чи «1» відповідно, як показано на рисунку 5. При цьому два поспіль знака «пробіл» в лінії зв'язку замінюються на один «пробіл» для зменшення інформаційної надлишковості.

x(t)	12	7	15	0	0	15	8	5	0	15	15	0	6	13
y(t)	-101-	-000-	1	0	0	1	-001-	-10-	0	1	1	0	-11-	-110-
Тег	{	H	Блок даних				B	(	Блок даних				)	}

Рисунок 5 – Лінійне кодування без шифрування машиною Тьюринга

Крипто-кодування структури потоку (шифрування тегів розмітки, але без шифрування блоків даних). Команди машини Тьюринга (теги розмітки) шифруються поточним ключом СЕК (рисунок 6), а блоки даних (від 4 біт) фрагментуються тегом «крапка» (код «-0-») для захисту від прямого перехоплення. На приймальній стороні дешифровані теги «крапка» видаляються. Швидкість крипто-кодування ключом СЕК наближається до швидкості нормального кодування по базовій таблиці команд. В даному ключі СЕК, тег «крапка» (номер 1) замінюється на тег номер 2 («кома» , шифр «-1-»).

СЕК	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	2	9	1	6	14	10	13	11	4	8	5	3	7	12
Код	1	010	0	11	111	011	110	100	01	001	10	00	000	101

x(t)	12	7	15	0	0	15	8	5	0	15	15	0	6	13
y(t)	-00-	-110-	1	0	-1-	0	1	-100-	-111-	0	1	-1-	1	0	-011-	-000-
Тег	;	}	10,01				]	$	01,10				[	H

Рисунок 6 – Крипто-кодування машиною Тьюринга (шифрування тегів)

Крипто-кодування/декодування кожного окремого символу цифрового потоку здійснюється прямою та зворотною парою логічних функцій «дешифратор-шифратор»: 4-бітовий двійковий код вхідного символу відображується у 16 вихідних логічних ліній, на одній з яких з'являється сигнал «1», а на всіх інших «0». Кожна вихідна лінія дешифратора підключена до свого шифратора, який має чотири логічні виходи для кодування 16-кового числа з чотирьох біт. Перетворення 4-бітового коду в трійкові лінійні сигнали машини Тьюринга нескладно реалізувати апаратно, наприклад на програмованих логічних інтегральних схемах (ПЛІС/FPGA), які здійснюють складні логічні операції за декілька машинних тактів (періодів кварцової частоти) в межах 0.01 мкс, тобто, практично, миттєво.

Описаний вище процес потокового крипто-кодування (шифрування структури цифрового потоку) настільки простий, що він практично не впливає на швидкодію передачі даних. Навіть без додаткового пост-шифрування даних, він реалізує достатній рівень захисту каналу передачі телеметрії в задачах дистанційного управління об'єктами мобільної ad-hoc мережі в реальному часі з точки зору можливого перехоплення зловмисником оперативного контролю над окремими об'єктами.

При цьому залишається можливість подальшого підвищення рівня захисту за рахунок додаткового пост-шифрування корисних та службових даних (незалежно і на додаток до інших наявних методів шифрування), у тому числі, з використанням процедури автентифікації. Окрім того, формальна граматика недетермінованої машини Тьюринга дозволяє конструювати велике різноманіття нових нестандартних протоколів крипто-кодування і шифрування «на випередження» (про-активно), тобто бути завжди на крок попереду, створюючи для противника нові виклики і вимушуючи його пасивно реагувати та наздоганяти лідера.

3) Мережева надлишковість.

Крипто-кодування створює певну мережеву надлишковість (додаткову службову інформацію), обсяг якої залежить від конкретного профілю протоколу, і впливає на показник інформаційної ефективності каналу зв'язку. Оцінимо вплив мережевої надлишковості методу крипто-стійкого динамічного кодування на інформаційну ефективність використання каналу порівняно з фреймами WiFi типу MPDU (з шифруванням даних) на розглянутому вище прикладі з урахуванням додаткової службової інформації на крипто-захист (див. таб. 1).

Вплив мережевої надлишковості захисту даних на інформаційну ефективність методу крипто-стійкого динамічного кодування

Для коректної порівняльної оцінки впливу надлишковості крипто-захисту на інформаційну ефективність каналу зв'язку для запропонованого в роботі методу крипто-стійкого динамічного кодування даних, розглянемо попередній приклад розрахунку інформаційної ефективності β по формулі (6), однак при цьому додатково врахуємо структурні компоненти крипто-захисту:

$$\beta = [Q_U + (OH_0 + OHC_0)] \ / [Q_U + (OH + OHC), \text{ де} \qquad (9)$$

Q_U – корисна інформація фрейму (однакова для порівняння); OH_0, OHC_0 – надлишкова інформація фрейму WiFi 802.11n типу MPDU покоління WiFi-4 (2009), обраного для порівняння відповідно без врахування та з урахованням полів крипто-захисту (див. таб. 1); OH та OHC – надлишкова інформація фрейму TMF у формулі (7) відповідно відповідно без врахування та з урахованням полів крипто-захисту WPA2. При цьому розглянемо аналогічні три варіанти інкапсуляції даних у стеку TCP/IP (див. рис. 4): а) спрощений (дані реального часу RTD вкладено безпосередньо в поле "Frame Body"; б) проміжний (RTD вкладено в сегмент UDP (Рис. 1); в) стандартний (RTD вкладено в поле "Payload" протоколу RTP, що працює разом з протоколом RTCP).

Метод крипто-стійкого динамічного кодування в режимі «шифрування структури фрейму» – тобто, шифрування тегів розмітки шляхом випадкового перемішування кодів команд машини Тьюринга (без додаткового пост-шифрування корисних та службових даних) – працює на рівні лінійного кодування практично так само, як і без крипто-захисту, за виключенням того, що блоки даних (від 4 біт і більше) фрагментуються тегом «крапка» (лінійний трійковий код тегу в базовій таблиці команд «-0-»). В процесі кодування, тег «крапка» («-0-») може змінитися на інший з набору 14 команд машини Тьюринга, що додає 3÷5 надлишкових трійкових символів (в середньому **4** символи на тег). За рахунок цього, блоки даних «маскуються» під теги (які в даному режимі шифруються). Тому хакер не може безпосередньо виділити блоки даних із цифрового потоку за правилом «чотири чи більше двійкових біт поспіль між двома синтаксичними знаками пробілу». На кожні 2 байти даних в режимі крипто-захисту додається **1** тег «крапка» (в середньому 4/2=2 трійкових символи на один байт). При цьому надлишкова інформація дорівнює $2 \cdot \log_2(3) \approx 2 \cdot 1.6$ біт ≈ 3.2 біт $\approx$ **0.4** байт на кожен байт корисних чи службових даних.

Зведемо в таблицю 3 необхідні числові параметри для порівняльної оцінки впливу надлишковості крипто-захисту на інформаційну ефективність каналу зв'язку по формулі (9): $\beta = [Q_U + (OH_0 + OHC_0)] / [Q_U + (OH + OHC)$.

Врахуємо, що обрана нами для порівняння технологія WiFi стандарту 802.11n (покоління WiFi-4, 2009) з фреймом типу MPDU, використовує протокол захисту WPA2 з шифром AES-128. Цей протокол також зворотно підтримується у наступних поколіннях від WiFi-5 (2013) до WiFi-7 (2024) за протоколом безпеки WPA3 (в режимі "WPA3 Personal"). При цьому додаткові поля на шифрування даних WPA2 складають **16** байт (див. Таб. 1):

– 8 байт AES CCMP header before data (Counter Mode CBC-MAC Protocol);

– 8 байт MIC after data (Message Integrity Code, або цифровий підпис).

Таблиця 3 – Порівняльні параметри надлишковості крипто-захисту

Q_U (байт)	1	2	4	8	16	32	64	128
OH_0	24/32/48	24/32/48	24/32/48	24/32/48	24/32/48	24/32/48	24/32/48	24/32/48
OHC_0	16	16	16	16	16	16	16	16
Σ_0 (байт)	**41/49/65**	**42/50/66**	**44/52/68**	**48/56/72**	**56/64/80**	**72/80/96**	**104/112/128**	**168/176/192**
OH	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4
OHC	0.4	0.8	1.6	3.2	6.4	12.8	25.6	51.2
Σ (байт)	**6.8**	**7.2**	**8.0**	**9.6**	**12.8**	**19.2**	**32**	**57.6**
$\beta_1 = \Sigma_0 / \Sigma$	6.0	5.8	5.5	5	4.4	3.8	3.3	2.9
$\beta_2 = \Sigma_0 / \Sigma$	7.2	6.9	6.5	5.8	5.0	4.2	3.5	3.0
$\beta_3 = \Sigma_0 / \Sigma$	9.6	9.2	8.5	7.5	6.3	5.0	4.0	3.3

Нижче на рисунку 7 представлено гістограму і параметричне сімейство функцій для порівняльної оцінки впливу надлишковості крипто-захисту на інформаційну ефективність β каналу зв'язку по формулі (9) відповідно до числових значень в таблиці 3. Горизонтальна вісь Q_U представлена в нелінійному масштабі для компактності представлення діапазону [1, 128] байт.

Порівняльний аналіз інформаційної ефективності засвідчує наступне.

1. В режимі дистанційного управління безпілотним об'єктом мобільної ad-hoc мережі в реальному часі (наприклад, FPV-дроном по радіоканалу) при великій ймовірності кіберзагроз, трафік телеметрії доцільно передавати на обґрунтовано-достатньому рівні кіберзахисту з метою мінімізації надлишковості мережевих протоколів та підвищення інформаційної ефективності каналу зв'язку.

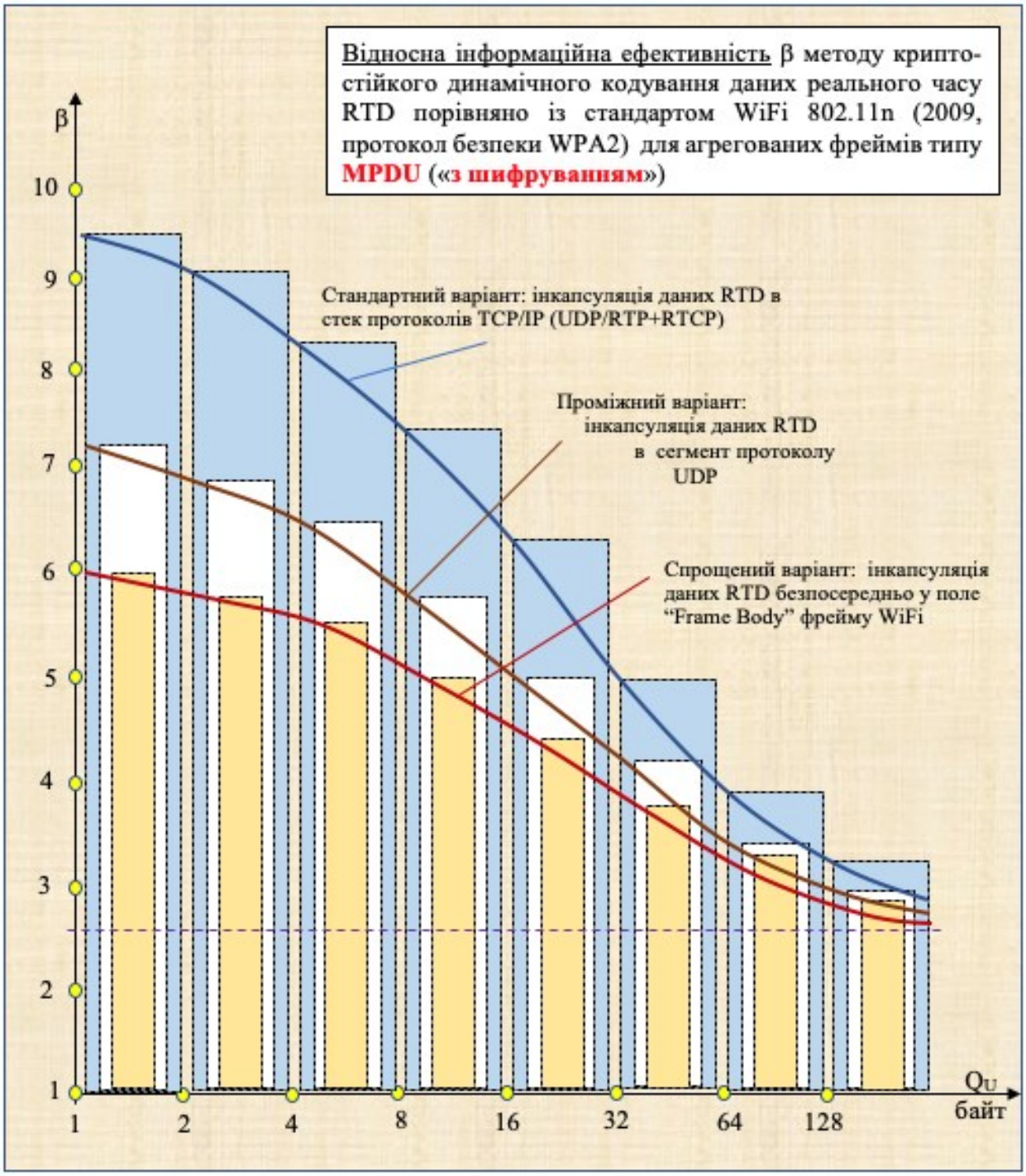


Рисунок 7 – Гістограма ефективності динамічного крипто-кодування

2. Для такого класу завдань кіберзахисту, запропонований в дисертації метод крипто-стійкого динамічного кодування даних, забезпечує *достатню стійкість* до зламу трафіка телеметрії на основі секретного поточного коду CEK (Current Encryption Code) для шифрування тегів розмітки, навіть без додаткового пост-шифрування корисних та службових даних фрейму тимчасовим секретним ключом TEK (див. розділ 2.3 дисертації).

Як показано вище, найсучасніші конвеєрні мікропроцесори (типу Core i9, Ryzen 9, Apple M3 тощо) з тактовою частотою f_{clc}=3÷5 ГГц потребують в середньому не менше 15 хвилин на злам поточного ключа крипто-кодування CEK. Натомість, поточне значення ключа CEK випадково змінюється приблизно через кожні 0.1 секунди, і жодного разу не повторюється протягом однієї місії мобільної ad-hoc мережі. При цьому, окремі місії не пов'язані між собою, кожна з них може тривати до 242 годин (біля 10 діб), а в межах однієї місії використовується лише 0.01% відсоток від кількості різних секретних ключів CEK. Остання дорівнює числу перестановок чотирнадцяти команд недетермінованої машини Тьюринга (NdTM): P(14)=14! = 87`178`291`200. Очевидно, що злам каналу телеметрії із затримкою на 15 хвилин є неактуальним, і його практично неможливо використати для оперативного впливу на мобільний об'єкт (наприклад, перехопити контроль над ним).

3. Метод крипто-стійкого динамічного кодування даних практично не вносить додаткової затримки у часі, оскільки потокове крито-кодування зі *змінним* ключом CEK виконується так само як і «просте кодування» з *незмінним* ключом (наприклад, базовим набором команд машини Тьюринга). Незначущу затримку обумовлює вставка тегів «крапка» що маскують блоки даних фрейму під «псевдо-команди» для унеможливлення їх прямого перехоплення на фізичному рівні каналу зв'язку (див. параметр OHC в таблиці 3 додатку).

4. Вплив мережевої надлишковості крипто-стійкого динамічного кодування на інформаційну ефективність β каналу зв'язку відображено на рисунку 7. Порівняно із популярним стандартом WiFi 802.11n (2009, протокол безпеки WPA2), який використовується в режимі агрегації WiFi-фреймів типу MPDU («з шифруванням»), запропонований в дисертації метод має виграш за показником відносної інформаційної ефективності β від **2.5** до **9.6** разів – в залежності від розміру блоків корисних даних реального часу RTD, а також від способу інкапсуляції даних у фрейми WiFi (стандартний, проміжний та спрощений варіанти) – див. таб. 3 та рис. 7 додатку.

III. Ефективність алгоритму динамічного розподілу даних в ad-hoc мережі Інтернету речей

Алгоритм динамічного розподілу даних в ad-hoc мережі IoT є складовою частиною запропонованого в дисертаційній роботі методу крипто-стійкого динамічного кодування і розподілу даних мобільної ad-hoc мережі IoT (розділи 3.2 та 3.3 дисертації). Даний алгоритм призначений для оптимізації цифрових потоків мережі в реальному часі, і розглядається як один з дієвих способів підвищення загальної стійкості мережі до впливу різноманітних негативних факторів (вихід із ладу окремих об'єктів чи елементів, зовнішнє втручання, активна протидія засобів РЕБ тощо) за рахунок постійної підтримки максимально можливої *продуктивності* мережі у динамічно змінних ситуаціях. Зокрема, чим вище продуктивність мобільної ad-hoc мережі, тим більш частина її ресурсів може бути спрямована на підвищення захисту інформації.

Показником *продуктивності* мережі (критерієм оптимізації), зазвичай, є сумарний потік мережі F_{max}, а завдання його підвищення відомо як "MaxFlow Problem" (MFP). Основними критеріями ефективності MFP-алгоритмів є швидкодія (алгоритмічна складність) та складність практичної реалізації. Зокрема, швидкодія оптимального розподілу потоків мобільної ad-hoc мережі IoT в реальному часі є ключовим фактором стійкості і живучості мережі при виконанні складної місії в агресивному оточенні (наприклад, військової операції на ворожій території).

В основу алгоритму динамічного розподілу даних в ad-hoc мережі IoT покладено удосконалений автором метод рішення комбінаторних задач на графах, який дозволяє *прискорити* процес оптимального розподілу потоків даних між полюсами мережі за критерієм максимального сумарного потоку, а також залучити штучний інтелект до реалізації алгоритму.

Проаналізуємо швидкодію запропонованого в дисертації алгоритму динамічного розподілу даних порівняно з найбільш популярними і близькими за суттю аналогами, різноманіття яких систематизовано в роботах [8] (1988), [9] (2023).

1) Алгоритм Форда-Фалкерсона (FFA, 1956) для орієнтованого зваженого планарного двополюсного ST-графу з полюсами S (виток), T (сток) має обчислювальну складність O(F·E), де O – нотація функції складності по кількості елементарних базових операцій (або часу на виконання алгоритму) F – максимальний потік від S до T; E – кількість дуг графу.

для ST-графу: $O(V^2 \cdot E)$, де V - кількість вершин графу; E - кількість дуг.

2) Edmonds-Karp Algorithm (EKA, [10], 1969; [11], 1972): $O(V \cdot E^2)$.

3) Dinitz Algorithm (DA, [12], 1970).

4) Goldberg-Tarjan Algorithm (GTA, 1986), [8]): $O[V \cdot E \cdot \log_2(V^2/E)]$.

5) Goldberg-Rao Algorithm (GRA, 1998, [12]): O(V·E). Ваги робер – бінарні числа (0 та 1).

6) Goldberg-Rao-Papp Algorithm (GRPA, 2006, [13]): O(V·E). Ваги робер – цілі числа 1÷U. $O\{[\min(V^{1/2}, E^{2/3},)] \cdot E \cdot \log_2(V^2/E) \cdot \log_2 U\}$, U– максим. вага ребра.

В публікації [8] зазначено, що всі відомі на той час ефективні Maxflow-алгоритми (MFP) працювали за принципом пошуку додаткового ST-шляху на орієнтованому зваженому планарному ST-графі, який збільшує потік (augmenting path). Ці алгоритми автори поділяють на два типи:

а) на кожній ітерації знаходився тільки один шлях (як в оригінальному алгоритмі Форда-Фалкерсона FFA); наприклад, алгоритм Едмондса-Карпа (1969), складність $O(V \cdot E^2)$. Для V=8, E=3·(V–2)=18, маємо $O(8 \cdot 18^2)=2592$

б) на кожній ітерації знаходилися усі найкоротші шляхи за кожну ітерацію (використовуючи багатошаровий мережевий підхід Дініца DA); наприклад, алгоритм Дініца (1970), складність $O(V^2 \cdot E) = O(8^2 \cdot 18) = 1152$.

Thus, by the start of the 1980s, it was known that MFP could be solved in either $O(V^3)$ or $O(E \cdot V \log_2 V)$ time, whichever is the faster ([9], c. 169 (2023).

При E=3·(V–2), V=8→: $O(E \cdot V \log_2 V)=3 \cdot (8-2) \cdot V \cdot \log_2 8=18 \cdot 8 \cdot 3= 432 < V^3 = 8^3= 512$.

(див. D. Sleator, 1980; D. Sleator, R. Tarjan, 1980).

Натомість, A. Goldberg і R. Tarjan у згаданій вище публікації запропонували “preflow-algorithm (GTA), де потік може «накопичуватися» у проміжних вузлах мережі); складність $O(V \cdot E \cdot \log_2(V^2/E))$. Для V=8, E=18 маємо $O(8 \cdot 18 \cdot \log_2(8^2/18))= =O(144 \cdot \log_2(64/18))=O(144 \cdot \log_2(3.56)) = O(144 \cdot 1.85) \approx 266$ (Див. [8]).

Наступні віхи. Обгрунтування можливості досягти теоретичного бар'єру мінімальної складності MFP-алгоритму – так званий "O(n·m) barrier" –, де алгоритмічна складність наближається до O(V·E). Наприклад, для V=8, E=18 цей бар'єр становить: O(V·E)=O(8·18) = 144 ([9], 2023).

Goldberg-Rao алгоритм (GRA, 1998) – умовно-поліноміальний (через невизначеність параметру U – максимальна ємність дуги на ST-графі) алгоритм складності $O\{E\cdot\min(V^{2/3}, E^{1/2})\cdot\log_2(V^2/E)\cdot\log_2 U$. Для V=8, E=9, U=4 (спрощений варіант нещільного графу) маємо: $O\{9\cdot\min(8^{2/3}, 9^{1/2})\cdot\log_2(8^2/9)\cdot\log_2 4$. Оскільки $8^{2/3} = 4 > 9^{1/2} = 3$, то $O(GRA)=O(9\cdot3\cdot\log_2 7.1\cdot2) \approx O(27\cdot2.8\cdot2)\approx 151$. Однак, даний результат досягається лише в окремому випадку нещільного графу та малими значеннями ваги дуг. Для щільних графів, даний алгоритм програє за швидкодію розглянутому вище алгоритму GTA (A. Goldberg, R.Tarjan).

Orlin алгоритм (Orlin, 2013): $O\{V\cdot E + E^{31/16}\cdot(\log_2 V)^2\}$ Ефективний лише розріджених графів з малим E. Однак, для щільного ST-графу: V=8, E=18 маємо: $O\{8\cdot18 + 18^{31/16}\cdot(\log_2 8)^2\} = O(144 + 243\cdot9\} \approx 144+2187 \approx 2331$.

Orlin-Gong алгоритм (OGA, 2021).

Алгоритмічна складність: $O\{(V\cdot E\cdot\log_2 V) / [\log_2(\log_2 V)+\log_2(E/V)]\}$.

Для щільного ST-графу: V=8, E=18 маємо:

$$O(OGA) = O\{(8\cdot18\cdot\log_2 8) / [\log_2(\log_2 8)+\log_2(18/8)]\} =$$

$$= O\{(144\cdot3)/[\log_2 3)+\log_2(2.25)]\} = O\{(432/(1.6+1.17) = O\{(432/2.77) \approx 156$$

Алгоритм динамічного розподілу даних (DDA).

Кількість основних ітерацій оцінюється розмірністю «дерева пошуку» найкоротших шляхів на вільно-орієнтованому повному планарному графі з V вершинами (див. Таб. 6 дисертації, с. 124, формула 12). Вона визначає теоретично мінімально можливу складність даного алгоритму при обчисленні кожного шляху за 2÷3 машинних такта на інтегральних схемах ПЛІС (FPGA):

$$O_T(DDA) \geq O_T\{2[(V-2)^2 +1]\};\ V \geq 3. \qquad (10)$$

Далі на рисунках 8 та 9 наведено детерміновані алгоритми за останні півстоліття ([8], 1988; [9], 2023, Додаток Е). Рисунок 10 демонструє статистичні алгоритми ([9], Додаток Е).

TABLE I. POLYNOMIAL-TIME ALGORITHMS FOR THE MAXIMUM FLOW PROBLEM*

Algorithm no.	Date	Discoverer	Running time	References
1	1969	Edmonds and Karp	$O(nm^2)$	[5]
2	1970	Dinic	$O(n^2m)$	[4]
3	1974	Karzanov	$O(n^3)$	[18]
4	1977	Cherkasky	$O(n^2m^{1/2})$	[3]
5	1978	Malhotra, Pramodh Kumar, and Maheshwari	$O(n^3)$	[21]
6	1978	Galil	$O(n^{5/3}m^{2/3})$	[11]
7	1978	Galil and Naamad; Shiloach	$O(nm(\log n)^2)$	[12, 25]
8	1980	Sleator and Tarjan	$O(nm \log n)$	[27, 28]
9	1982	Shiloach and Vishkin	$O(n^3)$	[26]
10	1983	Gabow	$O(nm \log U)$	[10]
11	1984	Tarjan	$O(n^3)$	[31]
12	1985	Goldberg	$O(n^3)$	[14]
13	1986	Goldberg and Tarjan	$O(nm \log(n^2/m))$	[16, 15]
14	1986	Ahuja and Orlin	$O(nm + n^2 \log U)$	[1]

* Algorithm 13 is presented in this paper.

Рисунок 8 – Поліноміальні MFP-алгоритми

Позначки: n – кількість вершин графу (V); m – кількість ребер (E);
U – максимальна ємність дуги ST-графу

Year	Bound	Reference
1962	$O(mnU)$	Ford and Fulkerson 28
1969	$O(m^2n)$	Edmonds and Karp 25, 26
1969	$O(m^2 \log U)$	Edmonds and Karp 25, 26
1970	$O(mn^2)$	Dinic 24
1970	$O(mn \log U)$	Dinic 24
1974	$O(n^3)$	Karzanov 50
1977	$O(n^2\sqrt{m})$	Cherkassky 20
1978	$O(m^{2/3}n^{5/3})$	Galil 34, 35
1978	$O(mn \log^2 n)$	Shiloach 72, Galil and Naamad 36
1980	$O(mn \log n)$	Sleator and Tarjan 73, 74
1986	$O(mn \log(n^2/m))$	Goldberg and Tarjan 41, 43
1989	$O(mn + n^2 \log U)$	Ahuja and Orlin 5
1989	$O\left(mn + n^2\sqrt{\log U}\right)$	Ahuja et al. 6
1989	$O\left(mn \log\left(\frac{n\sqrt{\log U}}{m}\right)\right)$	Ahuja et al. 6
1990	$O(n^3/\log n)$	Cheriyan et al. 18, 19
1990	$O(mn + n^{8/3} \log n)$	Cheriyan and Hagerup 17, Alon 7
1992	$O(mn + n^{2+\epsilon})$	King et al. 52
1993	$O\left(mn \frac{\log n}{\log(m/n)} + n^{2+\epsilon}\right)$	Phillips and Westbrook 67, 68
1994	$O\left(\frac{mn \log n}{\log(m/(n \log n))}\right)$	King et al. 53
1998	$O(m \min\{n^{2/3}, m^{1/2}\} \log(n^2/m) \log U)$	Goldberg and Rao 40
2013	$O(mn + m^{31/16} \log^2 n)$	Orlin 65
2021	$O\left(\frac{mn \log n}{\log\log n + \log(m/n)}\right)$	Orlin and Gong 66

Рисунок 9 – Класичні детерміновані алгоритми MFP (2023).

Overview of recent randomized maximum flow algorithms.

Year	Bound	Reference
2008	$O\left(m^{3/2}\operatorname{polylog} n\ \log U\right)$	Daitch and Spielman 23
2014	$O\left(m\sqrt{n}\operatorname{polylog} n\ \log^2 U\right)$	Lee and Sidford 57
2016	$O\left(m^{10/7}\operatorname{polylog} n\ U^{1/7}\right)$	Madry 59
2020	$O\left(m^{11/8}\operatorname{polylog} n\ U^{1/4}\right)$	Liu and Sidford 58
2020	$O\left(m^{4/3+o(1)}\ U^{1/3}\right)$	Kathuria et al. 51
2021	$O\left(\left(m+n^{1.5}\right)\operatorname{polylog} n\ \log U\right)$	van den Brand et al. 12
2021	$O\left(m^{3/2-1/328}\operatorname{polylog} n\ \log U\right)$	Gao et al. 39
2022	$O\left(m^{3/2-1/58}\operatorname{polylog} n\ \log U\right)$	van den Brand et al. 11
2022	$O\left(m^{1+o(1)}\log U\right)$	Chen et al. 16

Рисунок 10 – Статистичні MFP-алгоритми (2023)

Впорядковані по датам найбільш значущі за швидкодією MFP-алгоритми представлено в таблиці 6 для повних планарних ST-графів з V вершинами.

Таблиця 6 – Порівняння детермінованих MFP-алгоритмів

№	Date	Algorithm	Complexity O(V)	V=6 E=12	8 18	12 30	15 39	20 54	Rem
1	1969	Edmonds-Karp	$V{\cdot}E^2$	864	2692	10800	22815	58320	Ref
2	1970	Diniz	$V^2{\cdot}E$	432	1152	4320	8775	21600	[65]
3	1986	Goldberg-Tarjan	$V{\cdot}E{\cdot}\log_2(V^2/E)$	114	264	815	1479	3120	[67]
4	2021	Orlin-Gong	$(V{\cdot}E{\cdot}\log_2 V)$ / $[\log_2(\log_2 V)+\log_2(E/V)]$	78.5 72	157 144	408 360	683 585	1317 1080	Ref
5	2025	Tikhonov	$3{\cdot}\{(V-2)^2+1\}$	51	111	303	510	975	Ref
Performance gain vs Orlin-Gong algorithm				**1.5**	**1.4**	**1.4**	**1.3**	**1.3**	

На рисунку 8 відображено гістограму для оцінки ефективності запропонованого в дисертації методу динамічного розподілу даних в ad-hoc мережі IoT (позначеного як “Applicant”) за критерієм обчислювальної складності на планарному ST-графі з кількістю вершин V= {6, 8, 12, 15, 20}. Порівняно із найбільш швидким на сьогодні алгоритмів (Orlin-Gong, 2021), заявлений алгоритм (Applicant”) має виграш 1.3÷1.5 в залежності від кількості вузлів мережі (вершин графу V). При цьому Applicant” перетинає межу O(V·E).

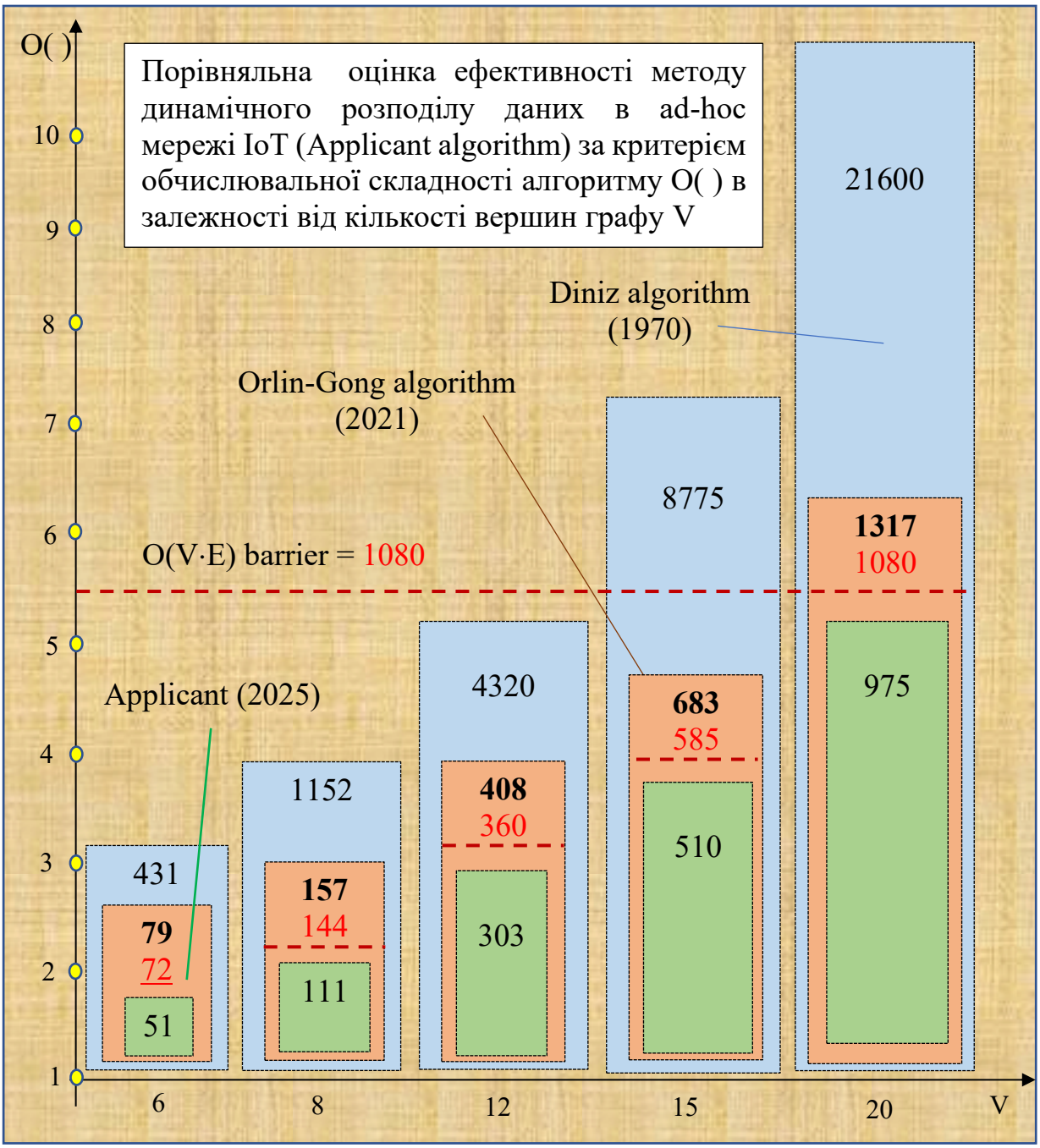


Рисунок 8 – Гістограма обчислювальної складності для алгоритмів розв’язання задачі про максимальний потік (MFP)

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ У ДОДАТКУ Е

1. Серков О.А. та ін. Теорія інформації та системи безпровідного зв'язку. *Інформаційно-керуючі системи на залізничному транспорті*. 2023. №1, с. 19-24. DOI: 10.18664/ikszt.v28i1.276339.
2. Rodriguas da Silva. Ethernet Timing Fundamentals: Understanding Preamble and Interframe Gap. URL: https://www.linkedin.com/pulse/ethernet-timing-fundamentals-understanding-preamble-gap-da-silva-cs1lf (visited: 03.01.2026).
3. 802.11 Frame Types and Formats. URL: 802.11 Frame Types and Formats (visited: 03.01.2026).
4. CWAP - MAC Header : Duration/ID. URL: https://mrncciew.com/2014/10/25/cwap-mac-header-durationid/ ((visited: 03.01.2026).
5. What Are Raw Sockets? URL: https://www.baeldung.com/cs/raw-sockets (visited: 03.01.2026).
6. Fault Tolerance vs High Availability: What's the Difference? URL: https://www.scalecomputing.com/resources/fault-tolerance-vs-high-availability (visited: 03.01.2026).
7. Understanding AES-128 encryption and its significance in the current threat landscape. URL: https://doverunner-com.translate.goog/blogs/understanding-aes-128-encryption-and-its-significance/?_x_tr_sl=en&_x_tr_tl=ru&_x_tr_hl=ru&_x_tr_pto=rq (visited: 03.01.2026).
8. Goldberg A.V., Tarjan R.E. A New Approach to the Maximum Flow Problem. *Journal of the Association for Computing Machinery*. 1988. V. 35, No. 4. P. 921-940. URL: https://www.cs.cmu.edu/afs/cs/academic/class/15499-s09/www/handouts/maxflow.pdf (visited: 03.01.2026).
9. Cruz-Mejía O., Letchford A.N. A survey on exact algorithms for the maximum flow and minimum-cost flow problems. *Networks*. 2023. V. 82, № 2. P. 167-176. DOI: https://doi.org/10.1002/net.22169Digital Object Identifier.
10. Edmonds J., Karp R.M. Theoretical improvements in algorithmic efficiency for network flow problems. *Proc. of Int. Conf. on Combinatorial Structures and their*

Applications (Calgary, 2-9 Lune, 1969). P. 248-264. DOI: https://doi.org/10.1145/321694.32169.

11. Edmonds J., Karp R. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *Journ. of Ass. for Computing Machinery*. 1972. V.19, №2. P. 248-264. URL: https://web.eecs.umich.edu/~pettie/matching/Edmonds-Karp-network-flow.pdf (visited: 03.01.2026).
12. Goldberg A., Rao S. beyond the flow decomposition barrier. *Journal ACM (JACM)*, V. 45, № 5.P. 783-797. DOI: https://doi.org/10.1145/290179.29018.
13. Papp D. The Goldberg–Rao Algorithm for the Maximum Flow Problem (2006). URL: https://www.cs.princeton.edu/courses/archive/fall06/cos528/handouts/Goldberg-Rao.pdf (visited: 03.01.2026).

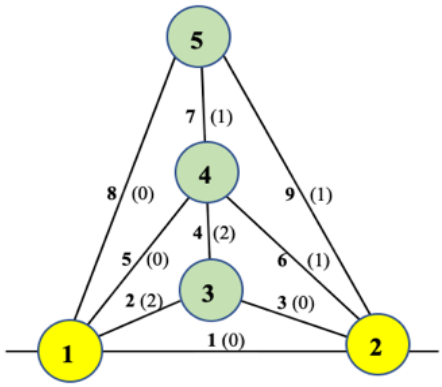
ДОДАТОК Є

МЕТОДИКА І ПРОГРАМНИЙ АЛГОРИТМ ОПТИМАЛЬНОГО РОЗПОДІЛУ ДАНИХ AD-HOC МЕРЕЖІ ЗА КРИТЕРІЄМ МАКСИМАЛЬНОГО ПОТОКУ

Розглядається побудова програмного алгоритму для оптимального розподілу даних мобільної планарної ad-hoc мережі за критерієм максимального сумарного потоку F на моделі нормалізованого зваженого вільно-орієнтованого планарного двополюсного ST-графу мережі з невід'ємними цілими вагами ребер і кількістю вершин V – далі позначеного як G(V) – так звана *зворотна задача оптимізації потоків* "Inverse MFP" (**I-MFP**) [63]. Для надійної верифікації алгоритму I-MFP розроблено алгоритм розв'язання прямої задачі оптимізації потоків "Direct MFP" (**D-MFP**) – побудова мінімального ST-графу (тобто, обчислення ваги усіх його ребер) для заданого розподілу максимального потоку F між полюсами графу. Алгоритм DT формулює повний набір еталонних варіантів розподілу потоків і забезпечує створення *ефективного генератора тестів* для алгоритму *зворотної (основної) задачі* I-MFP оптимального розподілу потоків даних між полюсами заданого ST-графу G(V).

Принципи побудови алгоритмів I-MFP та D-MFP демонструються на прикладі нормалізованого графу G(5) з 5 вершинами і 9 ребрами. Відповідні алгоритми для ST-графу G(V) з довільною обмеженою кількістю вершин V можуть бути побудовані за допомогою ШІ. Кількість незалежних шляхів ST-графу G(5): STP(V) = 2·(V-2), V ≥ 3 (див. формулу 10, рис. 3.5 дисертації).

Звідси STP(5) = **6**. З цих 6-ти шляхів 4 шляхи безальтернативні: S_1 ={1}; S_2 ={2, 3}; S_3 ={5, 6}; S_4 ={8, 9}. Шляхи S_5, S_6 є альтернативними: два 3-hop шляхи, або один 3-hop шлях + один 4-hop шлях. Кожен з альтернативних шляхів проходить через «вертикальні» ребра графу (4 та/або 7). При цьому, 3-hop потоки можуть проходити через пару вертикальних ребер тільки в протилежних напрямках (два варіанти: ↑↓ або ↓↑). Через кожне ребро потоки можуть проходити тільки в якомусь одному напрямку.



Також, можна показати, що алгоритм розв'язання зворотної задачі розподілу потоків (I-MFP) не залежить від величин цих потоків, і тому достатньо застосувати множину одиничних потоків. З урахуванням цього, побудуємо таблицю 1 для тестових варіантів розподілу одиничних потоків.

Таблиця 1– Варіанти розподілу **6**-ти одиничних потоків на ST-графі G(5)

F=6	Номери ребер ST-графу G(5)								
Тест **1**	1	2	3	4	5	6	7	8	9
S1÷S4	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**
S5	0	1	0	1	0	1	0	0	0
S6	0	0	0	0	0	1	1	1	0
T1: Edges	**1**	**2**	**1**	**1**	**1**	**3**	**1**	**2**	**1**
Тест **2**	1	2	3	4	5	6	7	8	9
S1÷S4	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**
S5	0	0	1	1	1	0	0	0	0
S6	0	0	0	0	1	0	1	0	1
T2: Edges	**1**	**1**	**2**	**1**	**3**	**1**	**1**	**1**	**2**
Тест **3**	1	2	3	4	5	6	7	8	9
S1÷S4	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**
S5	0	0	0	0	0	1	1	1	0
S6	0	0	1	1	0	0	1	1	0
T3: Edges	**1**	**1**	**2**	**1**	**1**	**2**	**2**	**3**	**1**
Тест **4**	1	2	3	4	5	6	7	8	9
S1÷S4	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**
S5	0	0	1	1	1	0	0	0	0
S6	0	0	1	1	0	0	1	1	0
T4: Edges	**1**	**1**	**3**	**2**	**2**	**1**	**1**	**2**	**1**
Тест **5**	1	2	3	4	5	6	7	8	9
S1÷S4	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**
S5	0	0	0	0	1	0	1	0	1
S6	0	1	0	1	0	0	1	0	1
T5: Edges	**1**	**2**	**1**	**1**	**2**	**1**	**2**	**1**	**3**
Тест **6**	1	2	3	4	5	6	7	8	9
S1÷S4	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**
S5	0	1	0	1	0	1	0	0	0
S6	0	1	0	1	0	0	1	0	1
T6: Edges	**1**	**3**	**1**	**2**	**1**	**2**	**1**	**1**	**2**

Результати розрахунків ребер для тестових графів представимо в таблиці 2.

Таблиця 2 – Тестові варіанти графу G(5) з **6**-ма одиничними потоками

F = 6	Шляхи потоків	Номери і ваги ребер ST-графу G(5)								
Тест №	(S1÷S4) + S5+S6	1	2	3	4	5	6	7	8	9
1	S5(3-hop): 2-4-6 S6(3-hop): 6-7-8	**1**	**2**	**1**	**1**	**1**	**3**	**1**	**2**	**1**
2	S5(3-hop): 3-4-5 S6(3-hop): 5-7-9	**1**	**1**	**2**	**1**	**3**	**1**	**1**	**1**	**2**
3	S5(3-hop): 6-7-8 S6(4-hop): 3-4-7-8	**1**	**1**	**2**	**1**	**1**	**2**	**2**	**3**	**1**
4	S5(3-hop): 3-4-5 S6(4-hop): 3-4-7-8	**1**	**1**	**3**	**2**	**2**	**1**	**1**	**2**	**1**
5	S5(3-hop): 5-7-9 S6(4-hop): 2-4-7-9	**1**	**2**	**1**	**1**	**2**	**1**	**2**	**1**	**3**
6	S5(3-hop): 2-4-6 S6(4-hop): 2-4-7-9	**1**	**3**	**1**	**2**	**1**	**2**	**1**	**1**	**2**

Програмний модуль I-MFP для ST-графу F(V), V∈[3, 4, 5]

Програмний модуль I-MFP для ST-графу F(V), V∈[3, 4, 5]

```
// Program module I-MFP (Inverse MaxFlow Problem)
V=5; print ("V =", V)
#E Nr: 1  2  3  4  5  6  7  8  9
E=[50, 1, 3, 1, 2, 1, 2, 1, 1, 2]     # E = Edges
print ("E =", E)
R=E; print ("R =", R)      # R = Rest edges
f1=R[1];  R[1]=0; print ("   f1=", f1, "  R[1]=",
R[1])
match V:
    case 3:
        f2=min([R[2], R[3]]); R[2]=R[2]-f2;
R[3]=R[3]-f2;
        print("   f2=", f2, "  R[2]=", R[2], "
R[3]=", R[3])
        F=f1+f2; print ("   F =", F)
        print ("   Rest edges R=", R)
    case 4:                  # 2-hop paths (f2, f3)
        f2=min(R[2], R[3]); R[2]=R[2]-f2; R[3]=R[3]-
f2;
        print("   f2=", f2, "  R[2]=", R[2], "
R[3]=", R[3])
        f3=min(R[5], R[6]); R[5]=R[5]-f3; R[6]=R[6]-
f3;
        print("   f3=", f3, "  R[5]=", R[5], "
R[6]=", R[6])
                        # 3-hop paths (f4)
        f4=  min(R[3],  R[4], R[5])
        f4a=min(R[2], R[4], R[6])
        print('      f4=', f4, 'f4a=', f4a)
        if(f4a>f4):
            f4=f4a
            R[2]=R[2]-f4; R[4]=R[4]-f4; R[6]=R[6]-f4
        else:
            R[5]=R[5]-f4; R[4]=R[4]-f4; R[3]=R[3]-f4
        F=f1+f2+f3+f4
        print('   f4=', f4)
        print ("   F=", F)
        print ("R=", R)
    case 5:                  # 2-hop paths (f2, f3,
f4)
        f2=min(R[2], R[3]); R[2]=R[2]-f2; R[3]=R[3]-
f2
        print("   f2=", f2, "  R[2]=", R[2], "
R[3]=", R[3])
        f3=min(R[5], R[6]); R[5]=R[5]-f3; R[6]=R[6]-
f3
        print("   f3=", f3, "  R[5]=", R[5], "
R[6]=", R[6])
        f4=min(R[8], R[9]); R[8]=R[8]-f4; R[9]=R[9]-
f4
        print("   f4=", f4, "  R[8]=", R[8], "
R[9]=", R[9])
        # 3-hop paths (f5, f6)
        f5= min(R[3], R[4], R[5])
        f5a=min(R[2], R[4], R[6])
        print('      f5=', f5, 'f5a=', f5a)
        if(f5a>f5):
            f5=f5a
            R[2]=R[2]-f5; R[4]=R[4]-f5; R[6]=R[6]-f5
        else:
            R[5]=R[5]-f5; R[4]=R[4]-f5; R[3]=R[3]-f5
        print('   f5=', f5)
        print('R=', R)
        f6 = min(R[6],  R[7], R[8])
        f6a=min(R[5], R[7], R[9])
        print('      f6=', f6, 'f6a=', f6a)
        if(f6a>f6):
            f6=f6a
            R[5]=R[5]-f6; R[7]=R[7]-f6; R[9]=R[9]-f6
        else:
            R[6]=R[6]-f6; R[7]=R[7]-f6; R[8]=R[8]-f6
        print('   f6=', f6)
        print('R=', R)
```

Test 1 (results)

V = 5
E = [50, 1, 3, 1, 2, 1, 2, 1, 1, 2]
R = [50, 1, 3, 1, 2, 1, 2, 1, 1, 2]
  f1= 1 R[1]= 0
  f2= 1 R[2]= 2 R[3]= 0
  f3= 1 R[5]= 0 R[6]= 1
  f4= 1 R[8]= 0 R[9]= 1
    f5= 0 f5a= 1
  f5= 1
R= [50, 0, 1, 0, 1, 0, 0, 1, 0, 1]
    f6= 0 f6a= 0
  f6= 0
R= [50, 0, 1, 0, 1, 0, 0, 1, 0, 1]
  f7= 0 f7a= 1
  f7= 1
R= [50, 0, 0, 0, 0, 0, 0, 0, 0, 0]
F= 6

Test 3 (results)

V = 5
E = [50, 1, 1, 2, 1, 1, 2, 2, 3, 1]
R = [50, 1, 1, 2, 1, 1, 2, 2, 3, 1]
  f1= 1 R[1]= 0
  f2= 1 R[2]= 0 R[3]= 1
  f3= 1 R[5]= 0 R[6]= 1
  f4= 1 R[8]= 2 R[9]= 0
    f5= 0 f5a= 0
  f5= 0
R= [50, 0, 0, 1, 1, 0, 1, 2, 2, 0]
    f6= 1 f6a= 0
  f6= 1
R= [50, 0, 0, 1, 1, 0, 0, 1, 1, 0]
  f7= 1 f7a= 0
  f7= 1
R= [50, 0, 0, 0, 0, 0, 0, 0, 0, 0]
F= 6

Test 6 (results)

V = 5
E = [50, 1, 3, 1, 2, 1, 2, 1, 1, 2]
R = [50, 1, 3, 1, 2, 1, 2, 1, 1, 2]
  f1= 1 R[1]= 0
  f2= 1 R[2]= 2 R[3]= 0
  f3= 1 R[5]= 0 R[6]= 1
  f4= 1 R[8]= 0 R[9]= 1
    f5= 0 f5a= 1
  f5= 1
R= [50, 0, 1, 0, 1, 0, 0, 1, 0, 1]
    f6= 0 f6a= 0
  f6= 0
R= [50, 0, 1, 0, 1, 0, 0, 1, 0, 1]
  f7= 0 f7a= 1
  f7= 1
R= [50, 0, 0, 0, 0, 0, 0, 0, 0, 0]
F= 6

Основні маркери коректності тестування програмного модуля I-MFP для оптимального розподілу потоків даних на нормалізованому планарному ST-графі G(5) з кількістю вершин від 3 до 5 полягають у наступному.

1) Кількість потоків $\{f_k\}$ має завжди дорівнювати $STP(V) = 2\cdot(V-2)$, $V \geq 3$.

2) Кожен тестовий потік має дорівнювати 1, а сумарний потік F має дорівнювати кількості потоків (напр., F≡ 6 для V=5).

3) Остаточний граф R завжди має бути нульовим (значення 50 у нульовому елементів R[0] масиву R є умовним щоб заповнити нульовий елемент масиву R, і в розрахунках цей елемент не використовується – для зручності нумерації шляхів і потоків починаючи від 1, оскільки в Python індексація масивів починається з 0).

V	3	4	5	6
STP(V)	2	4	6	8
F	3	4	6	8
R[] (final) ≡	0	0	0	0

4) Коректність алгоритму роботи модуля I-MFP не залежить від значення складових потоків (довільне невід'ємне ціле число 0, 1, 2, ...); тому алгоритм побудови тесту як прямої задачі D-MFP для оптимального розподілу потоків даних на основі одиничних потоків є достатнім для гарантованої верифікації алгоритму зворотної задачі I-MFP.

5) Повний тестовий набір для ST-графу з кількістю вершин від 3 до 5 містить 6 тестів (Таб. 2).

6) Побудова тестових наборів (а також обґрунтування необхідної кількості тестів набору) для ST-графів з кількістю вершин від 6 і більше є предметом подальших досліджень, у тому числі, в напрямку використання ШІ.