Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

Савка Максим Сергійович

УДК 621.382:004.056:004.738.5

ДИСЕРТАЦІЯ

«Енергоорієнтоване адаптивне керування криптографічним захистом Fog-вузлів IoT»

171 Електроніка

17 Електроніка та телекомунікації

Подається на здобуття наукового ступеня доктора філософії

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

Савка М.С.

Науковий керівник Пілінський Володимир Володимирович к.т.н., професор

Київ – 2026

АНОТАЦІЯ

Савка М.С. Енергоорієнтоване адаптивне керування криптографічним захистом Fog-вузлів IoT. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 171 «Електроніка». – Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», МОН України, Київ, 2026.

У дисертації одержано такі нові наукові результати:

Вперше запропоновано метод адаптивного енергоорієнтованого керування криптографічним навантаженням Fog-вузла, який, на відміну від наявних підходів, у процесі функціонування динамічно узгоджує вибір симетричного криптографічного примітиву з поточним рівнем енергозапасу апаратної платформи. Це дає змогу реалізувати концепцію керованої деградації: тимчасовий перехід до алгоритмів із меншою обчислювальною складністю розглядається не як зниження рівня безпеки, а як механізм забезпечення живучості системи та протидії атакам на виснаження батареї.

Набула подальшого розвитку концепція забезпечення апаратної живучості для мікропроцесорних систем мобільного IoT: запропоновано механізм, за яким у критичних енергетичних режимах система керування підтримує фізичну працездатність апаратного вузла внаслідок свідомого зниження обчислювального навантаження на мікропроцесор.

Вперше обґрунтовано використання інтегрального критерію якості J як аналітичного інструменту для формалізації компромісу між затримкою, енерговитратами та обчислювальними можливостями, що дає змогу математично підтвердити адекватність керування в досліджуваній електронній системі.

Удосконалено методологію апаратно-програмного моделювання для верифікації енергоорієнтованих систем на автономному вузлі з ARM-архітектурою. Удосконалення полягає в інтеграції розробленої емпіричної моделі енергоспоживання, що встановлює прямий аналітичний зв’язок між завантаженням CPU та струмами розряджання елементів живлення.

У дисертаційній роботі здійснено поглиблене дослідження апаратної живучості електронних Fog-вузлів мобільного Інтернету речей в умовах критичного обмеження електроживлення для удосконалення процесу проєктування вбудованих мікропроцесорних систем шляхом впровадження енергоорієнтованого адаптивного керування.

Метою дослідження є підвищення апаратної живучості та енергоефективності електронних систем Fog-вузлів мобільного IoT що працюють в умовах автономного електроживлення шляхом розроблення методу адаптивного керування процесорним навантаженням (зумовленим криптозахистом) на базі нечіткої логіки. Об'єктом дослідження є керування обчислювальними ресурсами та забезпечення фізичної живучості в апаратних вузлах туманних обчислень (Fog Computing) мобільного IoT, процеси енергоспоживання. Предметом дослідження є методи та моделі адаптивного керування режимами роботи мікропроцесорної системи із застосуванням нечіткої логіки для забезпечення енергоефективного балансу «безпека–енергія-затримка».

У роботі застосовано методи системного аналізу для формалізації енергетичних обмежень, методи теорії автоматичного керування (нечітка логіка) для динамічного вибору режимів навантаження, а також методи емпіричного й апаратно-програмного моделювання для калібрування сили струмів споживання на реальному фізичному пристрої.

Зміст дисертаційного дослідження викладений у чотирьох розділах, у яких подано результати теоретичних і експериментальних досліджень.

У вступі дисертаційної роботи обґрунтовано доцільність і актуальність обраного напряму досліджень. Показано, що в архітектурах мобільного IoT локальні апаратні вузли функціонують в умовах автономного електроживлення (резервне джерело, акумуляторна батарея). Водночас вони змушені застосовувати ресурсоємні алгоритми криптографічного захисту інформації, які створюють значне навантаження на електронні компоненти. За таких умов інтенсивні обчислювальні операції, особливо за відсутності виділених апаратних криптоприскорювачів, є головним споживачем енергії мікропроцесора. Це спричиняє швидке виснаження батареї та фізичну відмову устаткування. Зазначено потребу в розробленні систем керування, які

балансують процесорне навантаження та витрату енергії на операційному рівні електронного пристрою.

У першому розділі наведено результати критичного аналізу особливостей технічних рішень для побудови апаратних Fog-вузлів. Проаналізовано архітектуру вбудованих систем (System on a Chip (SoC)), обмеження автономних джерел електроживлення та вплив інтенсивних обчислювальних операцій на силу струму споживання мікропроцесора. З'ясовано, що фіксовані налаштування обчислювально складних алгоритмів є неефективними. Функціонування електронної компонентної бази в реальних умовах спричиняє динамічну зміну споживаної потужності, яка є прямо пропорційною тактовій частоті та квадрату напруги живлення. За високого мережевого навантаження ресурсоємні криптографічні перетворення експоненціально збільшують витрату енергії та ініціюють експоненціальне зростання паразитних струмів витоку в КМОН-транзисторах внаслідок нагрівання напівпровідникового кристала. Це явище формує ризик апаратного перезавантаження або спрацьовування механізмів термального тротлінгу (thermal throttling). З огляду на це доведено, що статичне застосування максимального захисту зумовлює нераціональне використання енергії та стрімке виснаження акумулятора. На завершення розділу сформульовано вимоги до енергоорієнтованого керування ресурсами мікропроцесора, обґрунтовано актуальність роботи та визначено завдання дослідження. Встановлено потребу реалізації стратегії контрольованої деградації, за якої зниження рівня захисту відбувається керовано, зберігаючи фізичну працездатність пристрою.

У другому розділі виконано теоретичне моделювання процесів адаптивного керування процесорним навантаженням. Fog-вузол формалізовано як керований об'єкт із обмеженими ресурсами. Вектор стану моделі поєднує енергетичні, обчислювальні та безпекові параметри (поточний заряд батареї, завантаження центрального процесора, інтенсивність трафіку та агрегований індикатор рівня загроз). Обґрунтовано застосування контролера на базі нечіткої логіки (Fuzzy Logic Controller), який динамічно змінює параметри роботи ядра залежно від стану системи.

Керувальний вихід моделі інтерпретують як вибір одного з дискретних криптографічних профілів: Safe, Survival, Balanced або Max.

Сформовано математичну модель, у якій введено інтегральний критерій якості для аналітичного оцінювання компромісу між продуктивністю системи та економією енергії. Функціонал агрегує три базові нормовані показники: затримку сервісу, енергетичні витрати та відхилення від цільового рівня безпеки. Застосування такого багатокритеріального підходу дає змогу математично збалансувати суперечливі вимоги щодо забезпечення високого рівня конфіденційності та необхідності збереження енергоресурсів пристрою. Доведено, що запропонований метод забезпечує керовану деградацію функцій на користь подовження часу роботи електронної апаратури: в умовах критичного дефіциту електроживлення система свідомо обмежує ресурсоємні операції (наприклад, переходить до полегшених алгоритмів хешування або шифрування), уникаючи повної відмови вузла.

Третій розділ містить опис практичної реалізації розробленої системи на базі контейнерної мікропрограмної архітектури. Обґрунтовано вибір мікрокомп'ютера Raspberry Pi 5 з багатоядерним ARM-процесором (Cortex-A76) як базової електронної платформи для створення дослідного апаратно-програмного стенда. Спроєктовано програмно-апаратний комплекс ASM-фреймворку (Adaptive Security Manager), який складається з ізольованих модулів: сервісу прикладного інтерфейсу (ASM API) та адаптивного агента безпеки (ASM-Agent). Побудовано емпіричну модель енергоспоживання, яка встановлює прямий зв'язок між обчислювальним навантаженням, завантаженням процесора та струмами розряджання.

У межах реалізації нечіткий контролер типу Такаґі-Сугено (Takagi-Sugeno) нульового порядку періодично зчитує системну телеметрію (рівень заряду, завантаження ядер) та показник рівня загрози, після чого формує узагальнений показник рівня безпеки. Для запобігання високочастотним перемиканням режимів (ефекту брязкоту) застосовано порогову дискретизацію та трапецієподібні функції належності. Відповідно до рішення контролера, система динамічно застосовує один із профілів захисту. Імітація криптографічного навантаження забезпечує відтворення пропорційної обчислювальної складності різних криптопримітивів (від еквівалента

AES-256-GCM до ChaCha20-Poly1305), маючи безпосередній вплив на тепловиділення та динамічну витрату енергії апаратної платформи.

У четвертому розділі проаналізовано результати експериментальних досліджень, виконаних із застосуванням методології Hardware-in-the-Loop (HiL) у реальних апаратних умовах. Проведено порівняльний аналіз динаміки розряджання батареї під керуванням нечіткої логіки (адаптивні стратегії) та в режимі фіксованого навантаження (статичні стратегії Fixed Max та Fixed Balanced). Для об'єктивності всі стратегії тестували за ідентичного часового профілю мережевого навантаження, що імітував фонову роботу, фазу масованої DDoS-атаки (високий рівень загрози) та фазу релаксації. Експерименти охоплювали сценарії повного забезпечення енергією (початковий заряд близько 100%) та критичного дефіциту (початковий заряд близько 35%).

За результатами апаратного-програмного моделювання з'ясовано, що у штатному сценарії за повного забезпечення енергією витрата заряду в адаптивному режимі становить 15.53% порівняно з 16.71% у статичному профілі Fixed Max. За умов критичного енергодефіциту адаптивна система демонструє найбільшу ефективність: витрата дорівнює 10.04% в адаптивному режимі проти 16.71% у статичному. Практичні вимірювання сили струмів споживання та навантаження процесора беззаперечно підтвердили висновки теоретичних розрахунків. У критичному сценарії контролер примусово знижує обчислювальне навантаження до профілів Survival або Balanced, стабілізуючи середнє завантаження процесора на безпечному рівні 31.06%. Це дає змогу уникнути перегріву напівпровідникового кристала, мінімізувати ризики відмов в обслуговуванні через переповнення апаратних буферів та подовжити час автономної роботи fog-вузла під мережевим навантаженням в 1.7 раза.

Застосування інтегрального критерію якості математично підтвердило адекватність роботи алгоритму керування: за наявності достатнього ресурсу стратегія мінімізує сумарний штраф, наближаючись до еталонних показників безпеки статичного профілю, а в разі енергодефіциту – безальтернативно надає пріоритет фізичному виживанню апаратного вузла. Окреслено межі застосування розробленої

системи. Встановлено, що відсутність виділених апаратних криптоприскорювачів на багатьох промислових платформах жорстко пов'язує безпекові алгоритми з енергоспоживанням процесора, що зумовлює потребу калібрування моделі для кожного нового типу SoC.

Отримані результати досліджень мають практичне значення і є прийнятними для використання під час проєктування та обслуговування вбудованих пристроїв IoT. Розроблені математичні та алгоритмічні моделі уможливлюють формування мікропрограмного забезпечення для розумних контролерів живлення, спрямованого на підтримання фізичної працездатності електроніки в умовах жорсткого дефіциту енергії. Наявність чітко розмежованих апаратних профілів завантаження створює основу для розроблення стійких до атак промислових шлюзів. Запропонована HiL-методика є ефективною для апаратного калібрування сили струмів та тепловиділення під час перенесення розробленого адаптивного фреймворку на інші промислові електронні платформи.

Ключові слова: вбудована система, мікроконтролер, Інтернет речей, периферійні пристрої, криптографічний захист, захист інформації, кібербезпека, енергоефективність, обчислювальне навантаження, надійність обладнання, нечітка множина, оцінка ефективності, моделювання, цілісність даних, керування ресурсами.

SUMMARY

Savka M.S. Energy-aware adaptive management of cryptographic security in IoT Fog nodes. – Qualifying scientific work on the rights of a manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 171 "Electronics". – National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ministry of Education and Science of Ukraine, Kyiv, 2026.

The following new scientific results were obtained in the dissertation:

For the first time, a method of adaptive energy-aware management of the cryptographic load of a Fog node is proposed, which, unlike existing approaches, dynamically aligns the choice of a symmetric cryptographic primitive with the current energy reserve level of the hardware platform during operation. This makes it possible to implement the concept of managed degradation: a temporary transition to algorithms with lower computational complexity is considered not as a decrease in the security level, but as a mechanism for ensuring system survivability and countering battery depletion attacks.

The concept of ensuring hardware survivability for microprocessor systems of the mobile IoT has been further developed: a mechanism is proposed by which, in critical energy modes, the control system maintains the physical operability of the hardware node as a result of a deliberate reduction of the computational load on the microprocessor.

For the first time, the use of an integral quality criterion as an analytical tool for formalizing the trade-off between latency, energy consumption, and computational capabilities is justified, which allows mathematically confirming the adequacy of management in the studied electronic system.

The methodology of hardware-software modeling for the verification of energy-aware systems on an autonomous node with ARM architecture has been improved. The improvement consists in the integration of the developed empirical power consumption model, which establishes a direct analytical relationship between CPU utilization and the discharge currents of battery cells.

The dissertation carries out an in-depth study of the hardware survivability of electronic Fog nodes of the mobile Internet of Things under critical power supply constraints

to improve the design process of embedded microprocessor systems through the implementation of energy-aware adaptive management.

The purpose of the study is to increase the hardware survivability and energy efficiency of electronic systems of mobile IoT Fog nodes operating under autonomous power supply conditions by developing a method of adaptive management of the processor load (caused by cryptographic protection) based on fuzzy logic. The object of the study is the computational resource management, and physical survivability provision in hardware nodes of fog computing of the mobile IoT, processes of energy consumption. The subject of the study is the methods and models of adaptive management of microprocessor system operating modes using fuzzy logic to ensure an energy-efficient "security-energy-delay" balance.

The work applies methods of system analysis to formalize energy constraints, methods of automatic control theory (fuzzy logic) for the dynamic selection of load modes, as well as methods of empirical and hardware-software modeling for calibrating power consumption currents on a real physical device.

The content of the dissertation research is presented in four chapters, which provide the results of theoretical and experimental studies.

The introduction justifies the relevance of the research, showing that local hardware nodes in mobile IoT architectures operate under autonomous power supply conditions. At the same time, they are forced to apply resource-intensive algorithms of cryptographic information protection, which create a significant load on electronic components. Under such conditions, intensive computational operations, especially in the absence of dedicated hardware cryptographic accelerators, are the main energy consumer of the microprocessor. This causes rapid battery depletion and physical equipment failure. The need to develop management systems that balance processor load and energy consumption at the operational level of the electronic device is noted.

The first chapter critically analyzes technical solutions for building hardware Fog nodes, including SoC architectures, autonomous power constraints, and the impact of intensive computations on microprocessor current consumption. It was found that fixed settings of computationally complex algorithms are inefficient. The functioning of the

electronic component base in real conditions causes a dynamic change in power consumption, which is directly proportional to the clock frequency and the square of the supply voltage. Under high network load, resource-intensive cryptographic transformations exponentially increase energy consumption and initiate an exponential growth of parasitic leakage currents in CMOS transistors due to the heating of the semiconductor crystal. This phenomenon forms the risk of a hardware reboot or the triggering of thermal throttling mechanisms. Given this, it is proven that the static application of maximum security leads to irrational use of energy and rapid battery depletion. At the end of the chapter, requirements for energy-aware management of microprocessor resources are formulated, the relevance of the work is justified, and the research tasks are defined. The need to implement a controlled degradation strategy is established, in which the reduction of the protection level occurs in a managed manner, preserving the physical operability of the device.

The second chapter performs theoretical modeling of the processes of adaptive processor load management. The Fog node is formalized as a controlled object with limited resources. The state vector of the model combines energy, computational, and security parameters (current battery charge, central processor utilization, traffic intensity, and an aggregated threat level indicator). The application of a controller based on fuzzy logic (Fuzzy Logic Controller), which dynamically changes the core operation parameters depending on the system state, is justified. The control output of the model is interpreted as the selection of one of the discrete cryptographic profiles: Safe, Survival, Balanced, or Max.

A mathematical model is formed in which an integral quality criterion is introduced for the analytical evaluation of the trade-off between system performance and energy savings. The functional aggregates three basic normalized indicators: service latency, energy costs, and deviation from the target security level. The application of such a multi-criteria approach allows mathematically balancing the conflicting requirements regarding the provision of a high level of confidentiality and the need to preserve the device's energy resources. It is proven that the proposed method ensures a managed degradation of functions in favor of extending the operation time of electronic equipment: under conditions of a critical power shortage, the system consciously limits resource-intensive operations (for

example, transitions to lightweight hashing or encryption algorithms), avoiding a complete failure of the node.

The third chapter contains a description of the practical implementation of the developed system based on a containerized firmware architecture. The choice of the Raspberry Pi 5 microcomputer with a multi-core ARM processor (Cortex-A76) as the basic electronic platform for creating an experimental hardware-software testbed is justified. The hardware-software complex of the ASM framework (Adaptive Security Manager) is designed, which consists of isolated modules: the application interface service (ASM API) and the adaptive security agent (ASM-Agent). An empirical power consumption model is built, which establishes a direct connection between computational load, processor utilization, and discharge currents.

Within the implementation, a zero-order Takagi-Sugeno fuzzy controller periodically reads system telemetry (charge level, core utilization) and the threat level indicator, after which it forms a generalized security level indicator. To prevent high-frequency mode switching (chattering effect), threshold discretization and trapezoidal membership functions are applied. According to the controller's decision, the system dynamically applies one of the protection profiles. The simulation of the cryptographic load ensures the reproduction of the proportional computational complexity of various cryptographic primitives (from the equivalent of AES-256-GCM to ChaCha20-Poly1305), having a direct impact on heat dissipation and the dynamic energy consumption of the hardware platform.

The fourth chapter analyzes the results of experimental studies performed using the Hardware-in-the-Loop (HiL) methodology in real hardware conditions. A comparative analysis of the battery discharge dynamics under fuzzy logic control (adaptive strategies) and in the fixed load mode (static strategies Fixed Max and Fixed Balanced) is carried out. For objectivity, all strategies were tested under an identical time profile of network load that simulated background operation, a massive DDoS attack phase (high threat level), and a relaxation phase. The experiments covered scenarios of full energy provision (initial charge of about 100%) and critical deficit (initial charge of about 35%).

Based on the results of hardware-software modeling, it was found that in a standard scenario with full energy provision, the charge consumption in the adaptive mode is 15.53%

compared to 16.71% in the static Fixed Max profile. Under conditions of a critical energy deficit, the adaptive system demonstrates the highest efficiency: consumption equals 10.04% in the adaptive mode versus 16.71% in the static one. Practical measurements of consumption currents and processor load unconditionally confirmed the conclusions of the theoretical calculations. In a critical scenario, the controller forcibly reduces the computational load to the Survival or Balanced profiles, stabilizing the average processor utilization at a safe level of 31.06%. This allows avoiding the overheating of the semiconductor crystal, minimizing the risks of denial of service due to hardware buffer overflows, and extending the autonomous operation time of the fog node under network load by 1.7 times.

The application of the integral quality criterion mathematically confirmed the adequacy of the control algorithm: in the presence of sufficient resources, the strategy minimizes the total penalty, approaching the reference security indicators of the static profile, and in the event of an energy deficit, it unconditionally prioritizes the physical survival of the hardware node. The limits of application of the developed system are outlined. It has been established that the absence of dedicated hardware cryptographic accelerators on many industrial platforms strictly links security algorithms with processor energy consumption, which necessitates calibrating the model for each new type of SoC.

The obtained research results have practical significance and are acceptable for use during the design and maintenance of embedded IoT devices. The developed mathematical and algorithmic models enable the formation of firmware for smart power controllers, aimed at maintaining the physical operability of electronics under conditions of a severe energy deficit. The availability of strictly demarcated hardware load profiles creates a basis for developing attack-resilient industrial gateways. The proposed HiL methodology is effective for the hardware calibration of currents and heat dissipation when transferring the developed adaptive framework to other industrial electronic platforms.

Keywords: embedded system, microcontroller, Internet of Things, edge devices, cryptographic protection, information protection, cybersecurity, energy efficiency, computational load, equipment reliability, fuzzy set, efficiency evaluation, modeling, data integrity, resource management.

Список публікацій здобувача:

1. Savka M. Analysis of the key models, methods, and means of data collection in the Internet of Things. *Technologies and Engineering*. 2025. Vol. 26, No. 2. P. 66–78. DOI: 10.30857/2786-5371.2025.2.6.
2. Савка М. Комплексна модель збору даних в мережах мобільного Інтернету речей. *Science-based technologies*. 2025. Т. 66, № 2. С. 226–237. DOI: 10.18372/2310-5461.66.20333.
3. Savka M., Pilinsky V. Adaptive energy-oriented hardware and software system for cryptographic protection of Internet of Things fog nodes based on fuzzy logic. *Technologies and Engineering*. 2025. Vol. 26, No. 5. P. 73–90. DOI: 10.30857/2786-5371.2025.5.6.

   Особистий внесок здобувача: розроблення архітектури спеціалізованого апаратно-програмного стенда на базі Raspberry Pi 5, імплементація взаємодіючих Docker-контейнерів, проведення експериментального тестування та написання основного тексту рукопису.

   Внесок Пілінського В.В.: участь у загальній постановці задачі дослідження, верифікація отриманих експериментальних результатів, контроль дотримання академічних стандартів викладу матеріалу та загальна редакція статті.
4. Савка М. С. Експериментальна оцінка енергоефективності захищених Fog-вузлів на базі Raspberry Pi. *Science, Technology and Culture: Interaction, Evolution and Progress* : матеріали міжнар. наук.-практ. конф. (Копенгаген, Данія, 21–23 груд. 2025 р.). Копенгаген, 2025. С. 16–20. ISBN 978-617-8680-29-9.

ЗМІСТ

ЗМІСТ 14

ВСТУП 20

1 АНАЛІЗ ПРОБЛЕМИ ТА ФОРМУВАННЯ ВИМОГ 28

1.1 Архітектури Інтернету Речей та сутність вузлів туманних обчислень 28

1.2 Вплив методів збору даних на енергоспоживання та апаратне завантаження вузлів Інтернету речей 33

1.3 Особливості мобільних систем Інтернету Речей 35

1.4 Електронні пристрої вузлів туманних обчислень та їх адаптація до умов експлуатації 37

1.5 Вимоги до механізму енергоорієнтованого криптозахисту вузла туманних обчислень 47

Висновки до розділу 51

2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ 53

2.1 Модель вузла туманних обчислень як об’єкта керування 53

2.2 Модель загроз та формування рівня загрози для вузла туманних обчислень 62

2.2.1 Класифікація загроз в аспекті впливу на електронні компоненти вузла 63

2.2.2 Навантажувальні атаки як тригер виснаження автономного джерела живлення 64

2.2.3 Формалізація агрегованого індикатора загроз 65

2.2.4 Джерела даних для формування рівня загрози 67

2.2.5 Зв’язок рівня загрози із вимогами щодо конфіденційності, цілісності, доступності та живучості вузла туманних обчислень 68

2.2.6 Рівень загроз у задачі керування 70

2.3 Модель якості криптографічного захисту 72

2.4 Інтегральний критерій якості та формулювання задачі вибору 80
2.4.1 Необхідність інтегрального критерію якості 80
2.4.2 Формальне введення інтегрального критерію якості 82
2.4.3 Нормалізація окремих показників затримки, енергії та безпеки 83
2.4.4 Вибір вагових коефіцієнтів для різних сценаріїв 85
2.4.5 Формулювання задачі адаптивного керування 87
2.4.6 Тип задачі та властивості розв'язку 88
2.4.7 Інтерпретація значень інтегрального критерію якості для порівняння стратегій 89
Висновки до розділу 90
3 МЕТОД ЕНЕРГООРІЄНТОВАНОГО АДАПТИВНОГО КЕРУВАННЯ 93
3.1 Концепція адаптивного фреймворку криптозахисту вузла туманних обчислень 93
3.1.1 Загальна архітектурна концепція фреймворку адаптивного керування на вузлі туманних обчислень 94
3.1.2 Контур моніторингу стану вузла та введення рівня загроз 96
3.1.3 Модель віртуальної батареї та облік навантаження центрального процесора 98
3.1.4 Інтеграція рівня загроз та адаптивного керування криптопрофілем 100
3.1.5 Застосування профілю 101
3.1.6 Контрольована деградація функціональності у реалізованому контурі 101
3.1.7 Логування та відтворюваність експериментів 102
3.2 Нечіткий контролер для вибору криптопрофілю 103
3.2.1 Місце нечіткого контролера в архітектурі фреймворку 103
3.2.2 Вхідні змінні нечіткого контролера 105
3.2.3 Лінгвістичні змінні та функції належності 105

3.2.4 Вихідна змінна рівня безпеки 107
3.2.5 База правил нечіткого контролера 108
3.2.6 Механізм запобігання брязкоту та порогова дискретизація рівня безпеки 109
3.2.7 Узгодження нечіткого контролера з інтегральним критерієм J 110
3.3 Криптографічні профілі та політики перемикання 111
3.4 Алгоритм реалізації та інтеграція у середовище туманних обчислень..... 120
3.4.1 Логічна структура реалізації на вузлі туманних обчислень..... 121
3.4.2 Алгоритм роботи агенту нечіткого контролера..... 123
3.4.3 Алгоритм роботи сервісу прикладного програмного інтерфейсу та зв’язок із профілями..... 124
3.4.4 Структури керуючих параметрів та конфігураційних матриць..... 126
3.4.5 Структура телеметрії та вимоги до відтворюваності 127
3.4.6 Інтеграція фреймворку в ширше середовище Інтернету Речей..... 128
Висновки до розділу 130
4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ, РЕЗУЛЬТАТИ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ 132
4.1 Апаратно-програмний стенд і методика експериментів 132
4.1.1 Hardware-in-the-Loop..... 132
4.1.2 Апаратна архітектура стенду..... 134
4.1.3 Мікропрограмна архітектура та ізоляція процесів тестового середовища 139
4.1.4 Сценарії експериментів..... 141
4.1.5 Метрики ефективності, збір даних та синхронізація логів..... 143
4.1.6 Оцінка похибок та обмежень експериментальної методики..... 145
4.1.7 Обґрунтування обсягу експериментальної вибірки..... 146
4.2 Результати експериментів..... 148

4.2.1 Енергоспоживання та автономність fog-вузла 148
4.2.2 Апаратне завантаження процесора 151
4.3 Порівняння зі статичними стратегіями 153
4.3.1 Постановка задачі порівняння та базові стратегії 153
4.3.2 Порівняння енергетичної ефективності 155
4.3.3 Порівняння навантаження CPU 159
4.3.4 Порівняння затримки оброблення запитів 161
4.3.5 Порівняння рівня безпеки 163
4.3.6 Аналіз компромісу через інтегральний критерій J 165
4.4 Робастність, чутливість, обмеження та рекомендації впровадження 169
4.4.1 Чутливість до вагових коефіцієнтів інтегрального критерію ***J*** 170
4.4.2 Чутливість до параметрів нечіткого контролера 172
4.4.3 Обмеження моделі батареї, оцінки Threat Level та апаратної платформи. 174
4.4.4 Стійкість до варіацій навантаження та шаблонів атак 176
4.4.5 Практичні рекомендації щодо вибору ваг і політик для різних умов 177
Висновки до розділу 180
ВИСНОВКИ 183
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 185
Додаток А 197
Додаток Б 199
Додаток В 200

Перелік позначень і скорочень

AEAD – Authenticated Encryption with Associated Data – автентифіковане шифрування з асоційованими даними

AES – Advanced Encryption Standard – симетричний алгоритм блочного шифрування

AES-GCM – Advanced Encryption Standard Galois/Counter Mode – режим лічильника з автентифікацією Галуа для блочного шифру

AMQP – Advanced Message Queuing Protocol – розширений протокол черг повідомлень

API – Application Programming Interface – прикладний програмний інтерфейс

ARM – Advanced RISC Machine – архітектура мікропроцесорів з удосконаленим скороченим набором команд

ASM – Adaptive Security Manager – адаптивний менеджер безпеки

CoAP – Constrained Application Protocol – протокол застосунків для обмежених пристроїв

CPU – Central Processing Unit – центральний процесор

DDS – Data Distribution Service – сервіс розповсюдження даних

DDoS – Distributed Denial of Service – розподілена атака на відмову в обслуговуванні

DTLS – Datagram Transport Layer Security – безпека транспортного рівня для датаграм

DVFS – Dynamic Voltage and Frequency Scaling – динамічне масштабування напруги та частоти

ECC – Elliptic Curve Cryptography – криптографія на еліптичних кривих

EDHOC – Ephemeral Diffie-Hellman Over COSE – протокол обміну ефемерними ключами Діффі-Геллмана

FLC – Fuzzy Logic Controller – нечіткий логічний контролер

HiL – Hardware-in-the-Loop – апаратне моделювання в контурі

HMAC – Hash-based Message Authentication Code – код автентифікації повідомлень на основі хешування

HTTP – HyperText Transfer Protocol – протокол передавання гіпертексту

IDS – Intrusion Detection Systems – системи виявлення вторгнень

IoT – Internet of Things – Інтернет Речей

LLN – Low-Power and Lossy Networks – мережі з низьким енергоспоживанням і втратами

MAC – Message Authentication Code – код автентифікації повідомлення

MQTT – Message Queuing Telemetry Transport – протокол передавання телеметрії з чергами повідомлень

OSCORE – Object Security for Constrained RESTful Environments – об'єктна безпека для обмежених RESTful середовищ

PUF – Physically Unclonable Function – фізично неклонована функція

QoS – Quality of Service – якість обслуговування

REST – Representational State Transfer – передавання стану представлення

RPS – Requests per Second – кількість запитів за секунду

RSA – Rivest-Shamir-Adleman – криптографічний алгоритм з відкритим ключем

SBC – Single-Board Computer – одноплатний комп’ютер

SoC – System on a Chip – складна електронна система, реалізована на одному кристалі

TLS – Transport Layer Security – безпека транспортного рівня

URLLC – Ultra-Reliable Low-Latency Communication – наднадійний зв'язок із низькою затримкою

V2X – Vehicle-to-Everything – комунікації між транспортним засобом та інфраструктурою

WSN – Wireless Sensor Networks – безпроводові сенсорні мережі

АЦП – аналого-цифровий перетворювач

ОЗП – оперативний запам'ятовувальний пристрій

ВСТУП

Обґрунтування вибору теми дослідження. У багатьох архітектурах мобільного Інтернету речей (Internet of Things, IoT) вузол туманних обчислень (Fog-вузол) є проміжною ланкою між периферійними пристроями та мережевими сервісами. З позиції електроніки такий вузол є вбудованою мікропроцесорною системою, переважно побудованою на базі архітектури системи на кристалі (System on a Chip (SoC)), що агрегує дані, виконує їх первинне оброблення та маршрутизацію. Роль локального обчислювального вузла зумовлює прямий вплив Fog-компонента на безперервність сервісу: його апаратна відмова або деградація обчислювальних потужностей здатна порушити передавання даних і керувальні контури [1]. Водночас такі пристрої мають виконувати ресурсоємні функції безпеки із застосуванням обмеженого апаратного устаткування, що зумовлює значне навантаження на їхні електронні компоненти [2].

Питання енергоорієнтованого криптографічного захисту ресурсно-обмежених вузлів IoT набуло значного поширення у фахових виданнях, індексованих у наукометричних базах Scopus та Web of Science. Зокрема, аналіз таксономії легковагових фреймворків у IEEE Internet of Things Journal за 2025 рік визначає адаптивні підходи як один із найбільш перспективних векторів розвитку галузі [3]. Питання енергетичного управління у туманних обчисленнях опрацьовують у межах систематизації ресурсів для застосунків Інтернету речей [4]. Комплексне оцінювання криптографічних примітивів для пристроїв з обмеженим енергетичним ресурсом підтверджує критичну потребу мінімізації обчислювальних витрат [5]. Проте більшість проаналізованих рішень мають статичний характер, що зумовлює потребу у розробленні гнучких методів керованої деградації (graceful degradation) захисту на основі безперервного моніторингу стану батареї.

У дисертації проаналізовано шляхи підвищення живучості (survivability) апаратних Fog-вузлів (IoT-шлюзів) за умов поєднання двох чинників: автономного електроживлення та високого обчислювального навантаження, зумовленого протидією кібератакам. У такій постановці пріоритетом є фізичне виживання

апаратного вузла: пристрій має залишатися функціональним достатньо довго, зберігаючи працездатність електронної платформи навіть під піковим інформаційним та обчислювальним навантаженням.

У практичних умовах експлуатації мобільні Fog-вузли функціонують в ситуаціях, коли електроживлення є нестабільним (резервне джерело, акумуляторна батарея, жорсткий ліміт енергії). За таких умов криптографічні операції, особливо за відсутності виділених апаратних криптоприскорювачів, перетворюються на одного з головних споживачів енергії мікропроцесора [6]. Тому вимога до живучості Fog-вузла набуває суто апаратного, енергоорієнтованого змісту: важливо керувати обчислювальним навантаженням так, щоб надмірні струми споживання не спричиняли стрімкого виснаження батареї до повної втрати працездатності системи.

Проблема є особливо гострою за інтенсивного трафіку, коли алгоритми захисту зумовлюють суттєве навантаження на центральний процесор й експоненціально збільшують витрату енергії. Застосування фіксованих протоколів у мобільному Fog-вузлі є обмеженим підходом: використання обчислювально складних алгоритмів за високого мережевого навантаження швидко виснажує елементи живлення. Це є нештатною ситуацією, коли інтенсивне оброблення інформаційних потоків зумовлює пікове споживання струму мікропроцесором. Відбувається комплексна взаємодія: інформаційне навантаження прямо конвертується у теплове та силове перевантаження електронної компонентної бази. Отже, постає потреба в системі керування, яка здатна балансувати апаратні ресурси «безпека–енергія-затримка» на операційному рівні електронного пристрою.

Цю проблематику додатково загострює реальна загроза цілеспрямованого виснаження батареї IoT-пристроїв унаслідок атак позбавлення сну [7]. Аналіз загроз, пов'язаних із живленням, та методів захисту від них наведено у сучасних оглядах стану питання [8]. Хоча відомі адаптивні протоколи дають змогу частково знизити енергоспоживання, вони переважно обмежуються зміною довжини ключів [9]. Заповнення цієї прогалини потребує розроблення апаратно верифікованого методу наскрізного керування криптографічним профілем Fog-вузла, який має забезпечувати витончену деградацію захисту від повноцінного режиму до безпечного вимкнення.

У дисертації запропоновано адаптивний вибір режиму роботи пристрою із застосуванням нечіткої логіки, що зумовлює зміну параметрів функціонування мікропроцесорного ядра відповідно до рівня заряду акумуляторної батареї та наявної зовнішньої загрози. Цей підхід реалізує інженерну концепцію керованого зниження продуктивності: у критичних режимах система зберігає фізичну працездатність апаратного вузла внаслідок керованої деградації криптографічних функцій, а за появи можливості – відновлює штатні обчислення. Для формалізації компромісу між затримкою, енергією та безпекою введено інтегральний критерій якості J (математичний інструмент, який кількісно оцінює баланс між витраченими апаратними ресурсами та рівнем кіберзахисту), яку застосовують для аналітичного оцінювання та кількісної верифікації системи керування.

Окремий акцент зроблено на перевірці підходу в умовах, близьких до реальних: набула розвитку методологія апаратно-програмного моделювання та створено дослідний стенд на базі платформи з ARM-архітектурою з емпіричною моделлю енергоспоживання. Ця модель встановлює прямий зв’язок «криптонавантаження ↔ завантаження CPU ↔ струми споживання / витрата енергії», що дає підстави вважати запропоноване енергоорієнтоване керування дієвим інструментом підвищення живучості електронної апаратури Fog-вузлів [10].

Мета і задачі дослідження. Метою дослідження є підвищення апаратної живучості та енергоефективності електронних систем Fog-вузлів мобільного IoT що працюють в умовах автономного електроживлення шляхом розроблення методу адаптивного керування процесорним навантаженням (зумовленим криптозахистом) на базі нечіткої логіки.

Для досягнення цієї мети у дисертації необхідно розв’язати такі завдання: 1) апаратні вимоги до життєздатності Fog-вузла; 2) енергоорієнтоване керування обчислювальними профілями; 3) калібрування та HiL-верифікацію підходу на реальному мікропроцесорному стенді.

Об’єкт дослідження – керування обчислювальними ресурсами та забезпечення фізичної живучості в апаратних вузлах туманних обчислень (Fog Computing) мобільного IoT, процеси енергоспоживання.

Предмет дослідження – методи та моделі адаптивного керування режимами роботи мікропроцесорної системи із застосуванням нечіткої логіки для забезпечення енергоефективного балансу «безпека–енергія». Для досягнення поставленої мети необхідно розв'язати такі завдання:

1. Виконати аналіз ролі Fog-вузла як апаратного IoT-шлюзу щодо живучості, обмежень елементів живлення та впливу зовнішніх обчислювальних запитів, а також сформувати вимоги до енергоорієнтованого керування ресурсами мікропроцесора.
2. Сформулювати постановку задачі адаптивного керування параметрами функціонування за змінних умов (поточний рівень заряду батареї та інтенсивність загрози) та визначити критерії оцінювання компромісу «безпека–затримка–енергія», включно із застосуванням комплексного показника ефективності (цільової функції).
3. Розробити метод адаптивного вибору режимів роботи із застосуванням нечіткої логіки з урахуванням принципу плавного зниження продуктивності для підтримання фізичної працездатності апаратного вузла за критичних режимів електроживлення.
4. Розробити мікропрограмну архітектуру динамічного перемикання обчислювальних профілів (Safe, Survival, Balanced, Max) у вбудованій платформі відповідно до керувального рішення.
5. Розробити та налаштувати програмно-апаратний стенд на базі ARM-процесора (Raspberry Pi 5) із зовнішнім генератором трафіку, що дасть змогу забезпечити збір даних щодо струмів споживання для верифікації керування.
6. Побудувати емпіричну модель енергоспоживання системи та застосувати її для аналізу впливу режимів роботи на інтенсивність розряджання батареї.
7. Виконати експериментальне порівняння адаптивного керування з фіксованим режимом максимальної продуктивності за репрезентативних сценаріїв (зокрема за повного заряду та за низького рівня заряду батареї), проаналізувати динаміку енерговитрат і сформувати висновки щодо адекватності підходу.

8. Окреслити межі застосування розробленої системи та визначити напрями подальшого розвитку експериментальної бази.

Методи дослідження. Для розв’язання поставлених завдань застосовано сукупність методів, притаманних розробленню електронних систем та мікропроцесорної техніки:

1. Методи системного аналізу для формалізації енергетичних обмежень та вимог до апаратної живучості вбудованих Fog-вузлів;
2. Методи теорії автоматичного керування (нечітка логіка) для динамічного вибору режимів навантаження залежно від рівня заряду батареї;
3. Методи емпіричного моделювання енергоспоживання на базі лінійної апроксимації розряджання батареї та прив'язки обчислювальної складності до архітектури ARM-ядра;
4. Апаратне моделювання для калібрування струмів споживання та перевірки розроблених алгоритмів на реальному фізичному вузлі Raspberry Pi 5.

Наукова новизна одержаних результатів полягає в такому:

1. Вперше запропоновано метод адаптивного енергоорієнтованого керування криптографічним навантаженням Fog-вузла, який, на відміну від наявних підходів, у процесі функціонування динамічно узгоджує вибір симетричного криптографічного примітиву з поточним рівнем енергозапасу апаратної платформи. Це дає змогу реалізувати концепцію керованої деградації: тимчасовий перехід до алгоритмів із меншою обчислювальною складністю розглядається не як зниження рівня безпеки, а як механізм забезпечення живучості системи та протидії атакам на виснаження батареї.
2. Набула подальшого розвитку концепція забезпечення апаратної живучості для мікропроцесорних систем мобільного IoT: запропоновано механізм, за яким у критичних енергетичних режимах система керування підтримує фізичну працездатність апаратного вузла внаслідок свідомого зниження обчислювального навантаження на мікропроцесор.
3. Вперше обґрунтовано використання інтегрального критерію якості J як аналітичного інструменту для формалізації компромісу між затримкою,

енергією та безпекою, що дає змогу математично підтвердити адекватність керування в досліджуваній електронній системі.

4. Удосконалено методологію апаратно-програмного моделювання для верифікації енергоорієнтованих систем на автономному вузлі з ARM-архітектурою. Удосконалення полягає в інтеграції розробленої емпіричної моделі енергоспоживання, що встановлює прямий аналітичний зв’язок між завантаженням CPU та струмами розряджання елементів живлення.

Особистий внесок здобувача. Особистий внесок здобувача полягає у виконанні повного циклу робіт із проєктування та дослідження системи керування:

1. Сформульовано апаратну проблему підвищення живучості вбудованих Fog-вузлів за умов жорсткого дефіциту електроживлення;
2. Розроблено метод адаптивного керування процесорним навантаженням на основі нечіткої логіки з урахуванням ємності батареї;
3. Синтезовано інтегральний критерій якості для підвищення енергоефективності мікропроцесорних систем;
4. Спроєктовано мікропрограмну архітектуру динамічного керування профілями навантаження;
5. Спроєктовано та фізично зібрано вимірювальний апаратно-програмний стенд на базі мікрокомп'ютера Raspberry Pi 5 для реєстрації струмів споживання та обчислювальної активності;
6. Виконано цикл натурних вимірювань, формалізовано обмеження методики (зокрема за відсутності апаратних криптоприскорювачів) та сформовано інженерні рекомендації.

Практичне значення одержаних результатів:

1. На апаратному стенді підтверджено ефективність розробленої системи керування: у штатному режимі («Adaptive Full Battery») витрата заряду становить 15,53 % порівняно з 16,71 % у статичному профілі, а у разі енергодефіциту (зниження заряду акумулятора до ≈ 35 %) алгоритм дає змогу знизити споживання до 10,04 % проти 16,71 %. Це зумовлює істотне підвищення живучості електронного вузла.

2. Запропонований метод енергоорієнтованого керування доцільно впровадити на рівні мікропрограмного забезпечення вбудованих пристроїв IoT для підтримання їхньої фізичної працездатності в умовах дефіциту енергії. Наявність чітко розмежованих профілів завантаження процесора (Safe, Survival, Balanced, Max) створює основу для проєктування розумних контролерів живлення.
3. Застосування HiL-методології дає змогу виконувати апаратне калібрування струмів споживання під час перенесення фреймворку на інші промислові електронні платформи та SoC-архітектури, що є важливим інструментом для розробників енергоефективної апаратури.

Апробація результатів дисертації. Основні результати дисертаційної роботи оприлюднено у 4 наукових роботах, у тому числі у 3 статтях фахових виданнях України категорії Б та 1 доповіді на міжнародних коференціях.

Публікації. Під час навчання в аспірантурі основні положення та результати дисертаційної роботи висвітлено в таких публікаціях:

1. M. Savka, «Analysis of the key models, methods, and means of data collection in the Internet of Things», Technologies and Engineering, вип. 26, вип. 2, с. 66–78, Лип 2025, DOI: 10.30857/2786-5371.2025.2.6.

2. М. Савка, «КОМПЛЕКСНА МОДЕЛЬ ЗБОРУ ДАНИХ В МЕРЕЖАХ МОБІЛЬНОГО ІНТЕРНЕТУ РЕЧЕЙ», Science-based technologies, вип. 66, вип. 2, с. 226–237, Лип 2025, DOI: 10.18372/2310-5461.66.20333.

3. M. SAVKA, V. PILINSKY, «ADAPTIVE ENERGY-ORIENTED HARDWARE AND SOFTWARE SYSTEM FOR CRYPTOGRAPHIC PROTECTION OF INTERNET OF THINGS FOG NODES BASED ON FUZZY LOGIC», Technologies and Engineering, вип. 26(5), с. 73–90, Жов 2025, DOI: 10.30857/2786-5371.2025.5.6.

4. Савка М. С. Експериментальна оцінка енергоефективності захищених Fog-вузлів на базі Raspberry Pi // Science, Technology and Culture: Interaction, Evolution and Progress: матеріали міжнар. наук.-практ. конф. (Копенгаген, Данія, 21–23 груд. 2025 р.). – Копенгаген, Данія, 2025. – ISBN 978-617-8680-29-9.

Структура та обсяг дисертації. Дисертація складається із вступу, чотирьох розділів, загальних висновків, списку джерел посилання і додатків. Повний обсяг роботи складає 200 сторінок, 26 рисунків, список використаних джерел із 101 найменувань на 12 сторінках, 3 додатки на 4 сторінках.

1 АНАЛІЗ ПРОБЛЕМИ ТА ФОРМУВАННЯ ВИМОГ

1.1 Архітектури Інтернету Речей та сутність вузлів туманних обчислень

Розвиток систем Інтернету Речей зумовив появу різноманітних архітектурних підходів, які визначають структуру взаємодії пристроїв, мережевої інфраструктури та обчислювальних ресурсів. Класичні архітектури IoT зазвичай визначають як трирівневі моделі, які містять рівень пристроїв (сенсорів та актуаторів), мережевий рівень передавання даних та рівень хмарних сервісів для зберігання й оброблення [11]. На рівні пристроїв система здійснює первинний збір даних із фізичного середовища із застосуванням сенсорів або виконання команд керування через актуатори, мережевий рівень забезпечує транспортування цих даних до віддалених обчислювальних центрів, а хмарний рівень виконує функції глибокої аналітики, зберігання та надання сервісів кінцевим користувачам. Хмарні центри даних, розташовані на значній відстані від джерел даних, не можуть забезпечити прийнятний рівень затримок для застосунків, які потребують миттєвої реакції – від промислових систем керування до автономних транспортних засобів [12].

Для подолання цих обмежень запропоновано розширені архітектури, які охоплюють проміжний рівень обчислень на межі мережі: граничні обчислення (edge computing) та туманні обчислення (fog computing). Граничні обчислення виконують оброблення даних безпосередньо на пристроях або найближчих мережевих вузлах (наприклад, базових станціях, точках доступу, маршрутизаторах), тоді як туманні обчислення є концептуально ширшим поняттям, що охоплює розподілені обчислювальні вузли з обмеженими, але суттєвими ресурсами, розташовані між пристроями та хмарою. Згідно з класичною таксономією fog computing, запропонованою в оглядах 2017–2018 років, fog-вузли описуються як міні-хмари на краю мережі, які реалізують функції локального оброблення, агрегації, фільтрації та попереднього аналізу даних перед їхнім передаванням у хмару або безпосередньо до кінцевих застосунків [13]. Отже, сучасну архітектуру IoT можна подати як модель Device–Fog–Cloud (рис. 1.1), де кожен рівень виконує специфічні функції: пристрої –

збір і первинне оброблення, fog-вузли – локальну аналітику й агрегацію, хмара – глобальне зберігання, інтелектуальний аналіз і надання сервісів.

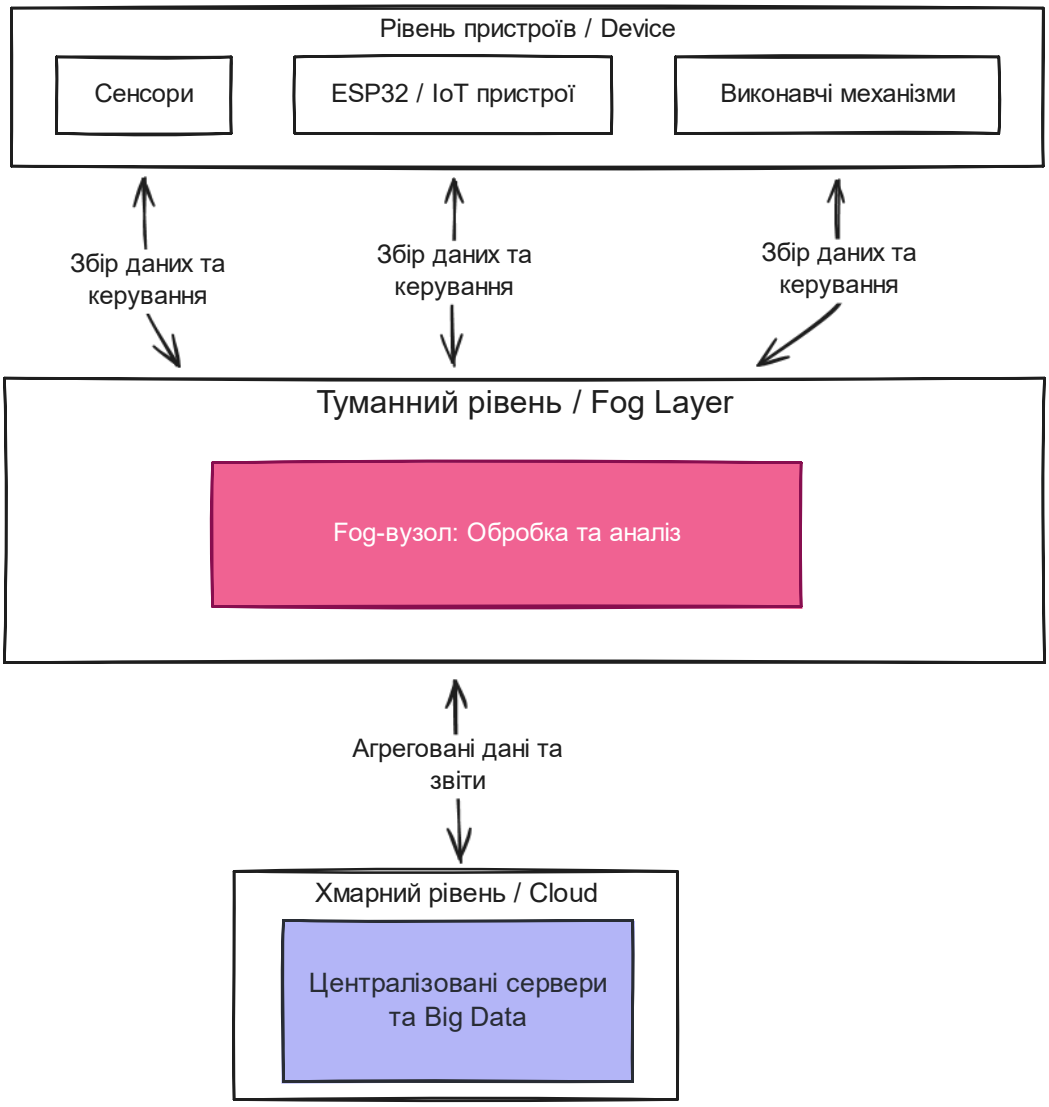

Рисунок 1.1 – Узагальнена трирівнева архітектура IoT

Впровадження fog-рівня в архітектуру IoT є природною еволюцією у напрямі зниження затримок, підвищення масштабованості та локальної автономності системи. Дослідження показують, що залучення fog та edge обчислень дає змогу суттєво зменшити час реакції розподілених IoT-систем порівняно з суто хмарними рішеннями: у координаційних середовищах типу cloud–fog–edge розміщення чутливих до затримок задач ближче до джерел даних (на fog та edge рівнях) зумовлює істотне скорочення end-to-end затримок та підвищення якості сервісу, що є критично

важливим для застосунків з жорсткими вимогами щодо затримок, включно з медичними системами моніторингу, промисловими системами керування та сервісами URLLC у 5G-мережах [14]. Крім того, fog-вузли здатні працювати в умовах нестабільного або періодичного зв'язку з хмарою, що забезпечує локальне прийняття рішень і буферизацію даних, та підвищує загальну доступність системи [15].

Fog-вузол можна визначити як електронний пристрій із обмеженими, але достатніми обчислювальними ресурсами, який розташований ближче до джерел даних, ніж хмарна інфраструктура, і здатний виконувати функції шлюза, агрегатора та локального обчислювального вузла одночасно. Типові апаратні реалізації fog-вузлів охоплюють одноплатні комп'ютери на базі архітектури «система на кристалі» (System on Chip), такі як Raspberry Pi різних поколінь, промислові шлюзи або спеціалізовані системи на кристалах (наприклад, Xilinx Zynq UltraScale+ MPSoC) [16]. Ці пристрої мають обмежену обчислювальну потужність процесора, оперативну пам'ять та енергоспоживання порівняно із серверами хмарних центрів даних, однак мають достатньо ресурсів для виконання завдань локального оброблення даних, запуску віртуалізованих сервісів у контейнерах (Docker, Kubernetes) та термінування захищених мережевих з'єднань. Експериментальні дослідження підтверджують, що fog-вузли на базі одноплатних комп’ютерів Raspberry Pi здатні обробляти потоки IoT-даних із незначним навантаженням на процесор та пам’ять, а це доводить їхню придатність для сценаріїв із великою кількістю сенсорів і обмеженими енергетичними ресурсами [17].

Функція fog-вузлів у збільшенні ефективності IoT-систем полягає насамперед у виконанні агрегації та попереднього оброблення даних. Fog-вузол приймає дані від множини IoT-пристроїв, виконує локальну фільтрацію шумових або незначущих вимірювань, агрегує дані за часовими вікнами або групами пристроїв, а також здійснює попередню аналітику, наприклад, виявлення аномалій або обчислення агрегованих статистик [18]. Процес оброблення на промислових майданчиках дає змогу значно скоротити обсяг даних, які передають у хмару: попередня агрегація, фільтрація й стиснення безпосередньо на fog-вузлах зумовлюють зменшення трафіку до частки від початкового обсягу, що спричиняє істотне зниження навантаження на

магістральні канали та вартості передавання [19]. Fog-вузли забезпечують можливість локального прийняття рішень у реальному часі без потреби звертання до хмари, що є необхідною умовою для застосунків із критичними вимогами щодо часу реакції, таких як промислові системи, транспортні системи або розумне енергопостачання [20].

Вимоги щодо доступності fog-вузла охоплюють високий час безперервної роботи, здатність працювати в умовах часткової або повної втрати зв'язку з хмарою та підтримку локального прийняття рішень. Моделювання доступності багаторівневих архітектур cloud–fog–edge доводить, що fog-рівень може суттєво підвищити загальну надійність системи внаслідок локального резервування сервісів і продовження функціонування навіть у разі відмов у хмарній інфраструктурі [21].

Fog-вузол також відіграє критичну роль у реалізації безпеки IoT-систем. На fog-рівні можливе виконання таких функцій захисту, як термінування захищених каналів зв'язку (наприклад, DTLS, TLS, MQTT over TLS), локальна автентифікація та авторизація IoT-пристроїв, виявлення аномалій у трафіку та поведінці пристроїв, а також шифрування та дешифрування даних перед їхнім передаванням у хмару або до кінцевих споживачів. Згідно з концепцією, яку наведено в NISTIR 8228, багато IoT-пристроїв мають обмежені можливості для реалізації повного спектра криптографічних механізмів і функцій безпеки через обмеження обчислювальної потужності та енергетичних ресурсів [22]. У таких ситуаціях fog-вузол може виступати як довірений посередник, який виконує ресурсоємні операції безпеки від імені обмежених пристроїв разом із забезпеченням end-to-end захисту даних. Наприклад, fog-вузли можуть реалізовувати шифрування та підпис на основі ідентифікаторів (identity-based encryption та identity-based signature) для автентифікації пристроїв та шифрування даних без потреби складної інфраструктури керування ключами на рівні кінцевих пристроїв, або використовувати блокчейн-підходи для забезпечення незмінності логів і прозорості операцій у мережі [23].

Водночас fog-вузол, під час виконання функції критичної точки довіри в архітектурі IoT, має бути об'єктом підвищених вимог щодо власної безпеки. Компрометація fog-вузла може спричинити втрату контролю над великою кількістю

пристроїв, витік конфіденційних даних або несанкціоновані зміни у фізичних системах, якими керують актуатори. Це вимагає реалізації на fog-вузлі механізмів захисту від несанкціонованого доступу, систем виявлення вторгнень (IDS), засобів контролю цілісності програмного забезпечення та регулярного оновлення політик безпеки. Порівняння характеристик кінцевого IoT-пристрою наведено на табл. 1.1.

Таблиця 1.1 – Порівняння характеристик кінцевого IoT-пристрою

Характеристика	Кінцевий IoT-пристрій (Device)	Туманний вузол (Fog Node)	Хмарний сервер (Cloud)
Обчислювальні ресурси	Низькі (мікроконтролери: MCU, ESP32, STM32)	Середні (одноплатні ПК: SBC, Raspberry Pi)	Дуже високі (центри обробки даних, ВМ)
Енергоспоживання	Мінімальне (робота від батарей)	Помірне (стаціонарне електроживлення)	Високе (потребує активного охолодження)
Типові функції безпеки	Апаратне шифрування (AES), автентифікація	Адаптивні політики (OPA/Rego), фільтрація трафіку	Глибокий аналіз загроз, ML-моделі безпеки
Латентність обробки	Мінімальна (реальний час, < 10 мс)	Низька (локальна мережа, 10–100 мс)	Висока (глобальна мережа, > 100 мс)
Зберігання даних	Тимчасове (буфер, КБ/МБ)	Короткострокове (кешування, ГБ)	Довгострокове (Big Data, ТБ/ПБ)

Разом із перевагами fog-обчислень, їхнє впровадження має низку обмежень і викликів. По-перше, проблема масштабування: велика кількість розподілених fog-вузлів ускладнює централізоване керування, розгортання політик безпеки та оновлення програмного забезпечення, особливо в гетерогенних середовищах, де fog-вузли можуть мати різні апаратні платформи, операційні системи та рівні довіри. По-друге, відсутність усталених стандартів для інтероперабельності fog-сервісів і протоколів обміну даними між fog-вузлами та хмарою ускладнює побудову масштабованих і мобільних рішень. По-третє, більшість сучасних архітектурних описів fog-систем зосереджені на забезпеченні високої продуктивності, належної якості обслуговування (QoS) і розподілі навантаження, тоді як питання енергоорієнтованого керування криптозахистом fog-вузла опрацьовано недостатньо.

Зокрема, у наявних роботах відсутні усталені підходи до адаптивного вибору криптографічних профілів залежно від стану енергетичних ресурсів fog-вузла (наприклад, рівня заряду акумулятора, навантаження на CPU) та поточного рівня загроз [24]. По-четверте, питання плавного зниження рівня захисту за критично низького заряду без повної відмови вузла практично не досліджені в контексті fog-вузлів для мобільних або енергообмежених IoT-сценаріїв.

1.2 Вплив методів збору даних на енергоспоживання та апаратне завантаження вузлів Інтернету речей

Збір даних в Інтернеті Речей визначають як базову функцію, що формує інтенсивність трафіку, часові характеристики та енергоспоживання [25]. В апаратному аспекті обрана модель збору даних диктує частоту виходу мікропроцесора з режиму сну та інтенсивність роботи радіомодуля, що є головними чинниками розряду автономного джерела електроживлення. В аналізі безпроводових сенсорних мереж (Wireless Sensor Networks, WSN) доведено, що вибір моделі безпосередньо впливає на енергетичну вартість забезпечення безпеки, оскільки визначає кількість переданих пакетів та місця виконання криптографічних операцій [26].

У фахових публікаціях моделі збору даних класифікують за топологією на централізовані, ієрархічні, децентралізовані та кластерні [27]. Ієрархічна модель та кластерні підходи дають змогу перенести частину обчислень на рівень туманних (fog) вузлів або головних вузлів кластера [28]. Однак у централізованих і однорангових архітектурах виконання повноцінних протоколів автентифікації потребує значних обчислювальних ресурсів для операцій з відкритим ключем, що критично перевантажує мікроконтролери та збільшує обсяг службового трафіку [29].

Застосування методів агрегації даних спрямовано на зменшення кількості переданих повідомлень і зниження активного часу роботи радіотрансивера [30]. Водночас перенесення попередньої аналітики на fog-рівень зменшує мережеве енергоспоживання, але зумовлює зростання завантаження центрального процесора [31]. Концепція захищеної агрегації даних суттєво збільшує обчислювальне

навантаження внаслідок використання складних схем гомоморфного шифрування [32]. Натомість оптимізовані схеми стиснення дають змогу зменшити енергоспоживання підсистеми зв'язку внаслідок скорочення кількості переданих бітів [33]. Для енергоорієнтованого захисту ключовим завданням стає розподіл операцій між кінцевим вузлом та fog-рівнем з урахуванням поточного стану батареї та завантаження процесора [34].

Режими організації потоків даних (періодичний, подієвий, орієнтований на запити) визначають характер навантаження на апаратні компоненти [35]. Інтеграція транспортних протоколів безпосередньо впливає на обсяг службового трафіку та час активності передавача [36].

Для IoT-вузлів із електроживленням від батареї збільшення кількості переданих байтів внаслідок надмірних криптографічних накладних витрат зумовлює зростання часу роботи радіомодуля, що впливає на ресурс батареї зіставно з додатковими обчисленнями на центральному процесорі [37]. Для fog-вузлів, що обслуговують значну кількість клієнтів, використання ресурсоємних протоколів автентифікації на базі сертифікатів перетворюється на основне джерело навантаження на CPU, що стимулює перехід до адаптивного налаштування рівня захисту [38]. Посилення криптографічного захисту неминуче збільшує обчислювальне навантаження та обсяг трафіку, що підвищує сумарне енергоспоживання [39]. Тому в концепції екологічного Інтернету речей (Green IoT) наголошують, що одночасне досягнення максимальної безпеки та мінімального енергоспоживання є неможливим, що вимагає багатокритеріального компромісу [40].

Топологія мережі, методи агрегації та вибір протоколів сукупно формують граничні умови для проєктування апаратних механізмів захисту [41]. У сценаріях інтенсивного трафіку fog-вузол перебуває в режимі майже безперервного виконання криптографічних операцій, що спричиняє швидке вичерпання енергії. Статичний вибір максимального профілю захисту для всіх режимів роботи зумовлює перевитрату енергетичних ресурсів. З огляду на це виникає потреба в розробленні електронних систем, які інтегрують поточний стан джерела електроживлення, завантаження CPU

та рівень загроз для адаптивного налаштування криптографічного навантаження в реальному часі [42].

Порівняння основних комунікаційних протоколів в аспекті накладних витрат та їхнього впливу на енергоспоживання електронних компонентів наведено у табл. 1.2.

Таблиця 1.2 – Порівняння основних протоколів прикладного рівня

Протокол	Модель взаємодії	Накладні витрати	Підтримка QoS	Механізми захисту	Вплив на енергоспоживання
MQTT	Publish–Subscribe	Низькі (від 2 байт)	Так (0, 1, 2)	TLS/SSL, MQTT-TLS	Помірний (TCP-сесія)
CoAP	Client–Server (REST)	Дуже низькі (від 4 байт)	Базова (CON/NON)	DTLS, OSCORE	Мінімальний (UDP-базований)
HTTP	Client–Server	Високі (десятки/сотні байт)	Немає (на рівні протоколу)	TLS/SSL (HTTPS)	Високий
AMQP	Pub-Sub / Queuing	Середні/Високі (від 8 байт)	Так (Settlement)	TLS/SSL	Високий
DDS	Data-centric Pub-Sub	Середні	Багаторівнева (20+ політик)	DDS Security (TLS/DTLS)	Помірний / Високий

1.3 Особливості мобільних систем Інтернету Речей

Мобільні системи Інтернету Речей формують архітектури, де частина кінцевих вузлів або шлюзів змінює просторове положення. Типові сценарії мобільного IoT охоплюють інтелектуальні транспортні системи, безпілотні літальні апарати та носимі медичні пристрої, що функціонують у гетерогенних мережах [43]. Естафетне передавання (handover) вимагає постійного перемикання радіомодулів, що регламентують відповідні стандарти для стільникового зв'язку [44]. Усе це зумовлює потребу в узгодженому керуванні криптографічними сесіями та ключовим матеріалом під час зміни точок доступу [45].

Змінна якість радіоканалу спричиняє часті розриви з'єднань і втрати пакетів, що в умовах масового машинного зв'язку створює жорсткі вимоги до апаратної частини

вузлів [46]. Традиційні протоколи захисту вимагають повторних процедур ініціалізації сеансу (handshake), що утримує радіотрансивер та центральний процесор в активному стані, виснажуючи автономне джерело електроживлення. Для ресурсно-обмежених платформ необхідним є застосування полегшених алгоритмів та мінімізація обчислювальних витрат на повторне встановлення захищених каналів.

Специфічні загрози мобільних систем пов'язані з фізичним доступом до устаткування, перехопленням трафіку та підміною базових станцій. Зловмисники мають змогу здійснювати фізичне втручання для вилучення криптографічних ключів або ін'єкції шкідливого коду безпосередньо в мікроконтролер.

Для забезпечення безперервної роботи адаптивних механізмів безпеки критично важливим є надійний моніторинг стану мережі та апаратних компонентів. Як показано у [47], ефективно вирішити це завдання дає змогу інтеграція мобільних систем із туманними (fog) обчисленнями та алгоритмами кластеризації. Така комплексна модель збору даних забезпечує просторово-часове впорядкування переривань мікропроцесора, зменшує кількість активних циклів трансивера та формує уніфікований потік телеметрії. Кешування параметрів (залишковий заряд батареї, завантаження CPU, потужність сигналу) на рівні fog-шлюзу створює передумови для подальшого керування живленням із застосуванням інструментів нечіткого висновку. Відповідно, запропонована архітектура збору даних стає фундаментом для побудови енергоефективного криптографічного фреймворку, здатного до адаптації в мінливих радіоумовах.

Туманний вузол у мобільному середовищі виконує роль проміжної апаратної платформи для кешування та локальної аналітики. В умовах автономного електроживлення тривале виконання складних криптографічних операцій зумовлює перегрів кристала процесора та швидку деградацію батареї. Наявні стандарти здебільшого не враховують стан енергоресурсів під час вибору алгоритмів шифрування. Це формує потребу в розробленні механізмів плавного зниження рівня захисту, які динамічно змінюють криптографічний профіль залежно від поточного заряду батареї, обчислювального навантаження та ідентифікованого рівня загроз.

Приклади мобільних сценаріїв та їхні апаратні обмеження наведено у табл. 1.3.

Таблиця 1.3 – Приклади мобільних IoT-сценаріїв

Сценарій	Типи вузлів (S: Sensors / F: Fog)	Характеристики мобільності	Радіотехнології (Physical/Link layer)	Ключові загрози (Апаратний та мережевий рівні)	Ресурсні обмеження (Electronics perspective)
Інтелектуальний транспорт (V2X)	S: Радари, LiDAR (CAN/Ethernet). F: OBU, RSU.	Висока динаміка (v≤150 км/год), ефект Допплера.	DSRC (802.11p), C-V2X (PC5/Uu), 5G NR V2X.	Handover attacks, Sybil-атаки, підміна RSU, Jamming.	Латентність (<10 мс), потужний DSP/NPU для відео.
Групове застосування UAV (Swarm)	S: IMU, RTK-GPS. F: Дрон-лідер, наземна станція (GCS).	3D-маневреність, швидка зміна топології графа мережі.	FHSS (завадостійкість), LoRa (телеметрія), Wi-Fi 6 (Video), 5G.	GPS-spoofing, Hijacking, атаки на протоколи MAVLink/UAV CAN.	Питома енергоємність (Wh/kg), обмеження на масу обчислювача.
Носимі пристрої (Healthcare)	S: Біосенсори. F: Смартфон, Edge-hub.	Низька швидкість (v≤5 км/год), переривчастий зв'язок.	Bluetooth LE (BLE), Zigbee, NFC, ANT+, IEEE 802.15.4.	Side-channel attacks, Eavesdropping, фізичне втручання.	Ultra-low power (uA у режимі сну), мініатюризація (SoC/SiP).
Мобільні польові платформи (Tactical)	S: Сенсори середовища. F: Мобільний командний пункт (Rugged).	Змінна мобільність, робота у зоні прямої та непрямої видимості (NLOS).	LoRaWAN (Long-range), MANET, приватні LTE, SATCOM.	Rogue Base Station, дешифрування трафіку, фізичний тамперінг.	Електромагнітна сумісність (EMC), автономність (Power Density).

1.4 Електронні пристрої вузлів туманних обчислень та їх адаптація до умов експлуатації

На відміну від суто програмних абстракцій fog/edge-платформи, електронний fog-вузол має фіксовану апаратну конфігурацію у вигляді системи на кристалі, модулів пам’яті, підсистеми живлення та мережевих інтерфейсів, що задає верхні межі доступних обчислювальних, енергетичних та теплових ресурсів для реалізації

механізмів захисту інформації. Fog-вузол є вузлом проміжного рівня із суттєво складнішою архітектурою порівняно із сенсорними вузлами, однак залишається значно обмеженішим за ресурсами порівняно із центрами оброблення даних, що посилює вимоги щодо енергоефективної реалізації криптографічних і протокольних засобів [48].

Функціональна роль електронного fog-вузла охоплює задачі локального оброблення потоків телеметрії, виконання сервісів агрегації подій, кешування, попередньої аналітики та реалізації локальних контролерів. У такому разі fog-вузол одночасно виступає точкою концентрації загроз і критичною точкою реалізації механізмів автентифікації, авторизації та захисту каналів зв'язку (рис. 1.2).

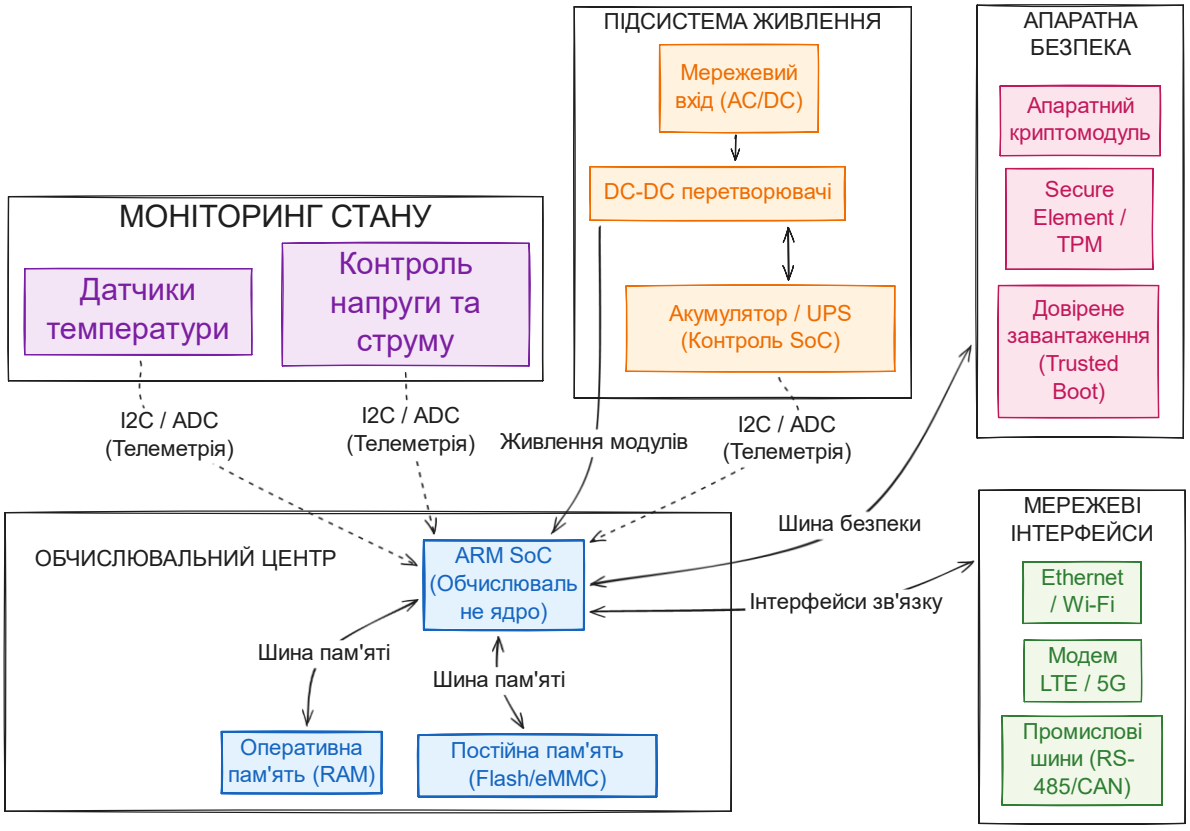

Рисунок 1.2 – Структурна схема електронного пристрою fog-вузла з виділенням модулів

У промислових сценаріях fog-вузол інтегрують у топологію, яка містить промислові шини (наприклад, Modbus/TCP, PROFINET) і безпроводові підсистеми (Wi-Fi, LTE/5G). Він також може виконувати роль шлюзу між доменами з різними політиками безпеки, що посилює вплив апаратних обмежень на вибір

криптографічних профілів і режимів функціонування. Саме апаратні межі щодо продуктивності процесора, обсягу оперативної пам’яті, доступної потужності електроживлення та допустимого тепловиділення визначають, які комбінації криптоалгоритмів, протоколів захисту та частот ротації ключів здатні функціонувати стабільно без ризику перевантаження вузла, надмірного нагрівання чи неконтрольованих перезавантажень.

Електронні платформи fog-вузлів поділяють на кілька характерних класів, серед яких одноплатні комп’ютери загального призначення (Single-Board Computers(SBC)), промислові шлюзи та контролери, а також спеціалізовані edge/fog-модулі на базі SoC з інтегрованими засобами безпеки. Одноплатні комп’ютери на кшталт Raspberry Pi 4/5 та аналогічних ARM-плат широко застосовують для прототипування та малих інсталяцій внаслідок поєднання відносно потужних багатоядерних ARM-процесорів, достатнього обсягу оперативної пам’яті, підтримки повноцінних Linux-дистрибутивів і розвиненого набору мережевих інтерфейсів (Gigabit Ethernet, дводіапазонний Wi-Fi, Bluetooth). У промислових застосуваннях, де важливими є підвищена надійність і розширений температурний діапазон, використовують промислові шлюзи на базі x86 або ARM, які мають захищені корпуси, широкий набір інтерфейсів (RS-485, CAN, промислові Ethernet-варіанти), варіанти живлення від постійної напруги та інтегровані модулі стеження за станом електроживлення. Спеціалізовані edge/fog-модулі містять SoC з інтегрованими безпроводовими інтерфейсами (Wi-Fi, LTE-категорії для міжмашинної взаємодії (Machine-to-Machine, (M2M)), 5G-модеми) та апаратними компонентами безпеки, такими як апаратні криптоакселератори для симетричної та асиметричної криптографії, модулі довіреного зберігання ключів і фізично неклоновані функції (Physically Unclonable Functions, (PUF)).

Типові характеристики зазначених платформ охоплюють діапазони тактових частот від енергоощадних сотень мегагерц для вбудованих SoC до декількох гігагерц для промислових x86-процесорів, обсяги оперативної пам’яті від сотень мегабайт до кількох гігабайт, наявність апаратних механізмів віртуалізації та криптоприскорення, а також підтримку широкого спектра дротових і безпроводових інтерфейсів. Широке впровадження апаратних модулів безпеки (Hardware Security Module (HSM), захищені

елементи, PUF) істотно впливає на архітектуру системи, оскільки переводить частину криптографічних операцій із загальних процесорних ядер на спеціалізовані блоки. Це знижує навантаження на CPU та енергоспоживання під час реалізації TLS, IPsec та інших протоколів. Гетерогенність fog-інфраструктури є однією з ключових проблем безпеки: у реальних розгортаннях одночасно використовують відносно слабкі ARM-вузли та потужніші промислові контролери, що ускладнює уніфікацію криптографічних профілів і політик захисту в масштабі всієї системи.

Енергоживлення електронних fog-вузлів забезпечують за кількома схемами, серед яких живлення від стаціонарної електромережі, від акумуляторів/батарей та гібридні конфігурації із застосуванням джерел безперебійного живлення (UPS) або інтегрованих DC-DC-перетворювачів для резервного та мобільного режимів. У промислових і енергетичних системах fog-вузли часто розміщують у віддалених або важкодоступних місцях, де є ймовірними тривалі відключення електроенергії, а тому блоки живлення містять резервні акумулятори або суперконденсатори для гарантування завершення критичних транзакцій чи коректного завершення роботи в разі зникнення мережевого електроживлення. Навіть за умови підключення до стаціонарної мережі питання енергоспоживання fog-вузла стає критичним унаслідок вимог щодо енергоефективності та обмежень на тепловиділення в компактних корпусах із пасивним охолодженням.

Для мобільних fog-вузлів, зокрема в транспортних, логістичних та польових сценаріях, енергоживлення часто ґрунтують на акумуляторних батареях, бортових мережах живлення транспортних засобів або комбінованих рішеннях із локальною генерацією, де стан заряду батареї та профіль споживання безпосередньо впливають на доступний обчислювальний та тепловий бюджет. За умови фіксованої ємності батареї суттєву частину енергоресурсу споживають саме обчислювальні завдання з інтенсивними операціями, включно з криптообробленням та шифруванням трафіку, що мотивує інтеграцію стану джерела електроживлення до політик безпеки [49]. Для вузлів, які працюють у режимі енергодефіциту, профіль навантаження та обрана конфігурація захисту здатні визначати момент вимушеного переходу до

деградованого режиму або повної зупинки сервісів, якщо не враховувати залишковий ресурс батареї під час проєктування механізмів захисту.

Механізми адаптації на рівні апаратури та системного програмного забезпечення для fog-вузлів ґрунтують на класичних засобах енергоменеджменту, до яких зараховують динамічне масштабування напруги та частоти (Dynamic Voltage and Frequency Scaling, (DVFS)), відключення невикористовуваних функціональних блоків, а також використання різних режимів сну та пробудження. DVFS дає змогу зменшувати напругу та тактову частоту процесора або обчислювальних ядер у разі зниження навантаження. У такий спосіб знижують динамічну складову споживаної потужності, а також обмежують пікове тепловиділення, що є ключовим для компактних ARM-базованих fog-плат. Для багатоядерних SoC, типових для SBC та промислових edge-платформ, застосовують техніки керування тактовою частотою окремих ядер або кластерів (clock gating, power gating), що дає змогу виводити невикористовувані ядра в низькопотужні стани без повної зупинки всієї системи.

Системне програмне забезпечення fog-вузлів, включно з повноцінними Linux-подібними дистрибутивами та операційними системами реального часу (Real-Time Operating System, (RTOS)), інтегрує драйвери та підсистеми енергоменеджменту, які використовують апаратні можливості DVFS, керування живленням доменів та інтерфейсів для підтримки стабільної роботи в умовах змінного навантаження й температурних коливань. Координація декількох політик через єдину систему моніторингу дає змогу динамічно змінювати частоту процесора, розподіл завдань і режими енергозбереження, з огляду як на продуктивність, так і на енергоспоживання та температурні обмеження [50]. Для малопотужних вбудованих систем застосовують полегшені схеми навчання для DVFS, які використовують спостережувані параметри навантаження для побудови моделей прогнозування та вибору оптимальних конфігурацій частоти й напруги, що є можливим для використання як основи для узгодження енергопрофілю з вимогами щодо захисту інформації.

Основні класи електронних платформ fog-вузлів наведено у табл. 1.4.

Таблиця 1.4 – Основні класи електронних платформ fog-вузлів

Клас платформи	Процесор (CPU)	Оперативна пам'ять (RAM)	Тип електроживлення	Апаратне криптоприскорення	Діапазон температур
Одноплатний комп'ютер (SBC)	RISC (ARM Cortex-A), 4–8 ядер. Обмежена підтримка віртуалізації (KVM/QEMU).	2–16 ГБ, LPDDR4/LPDDR5 (Non-ECC).	DC-джерело (USB-C, 5V/PoE). Акумулятор через модуль UPS.	Програмно-апаратне (інструкції AES у складі ARM NEON).	Стандартний (0…+50 °C).
Промисловий шлюз	x86 (Intel Atom/Core) або ARM. Повна підтримка апаратної віртуалізації (VT-x/AMD-V).	4–32 ГБ, DDR4/DDR5 (часто з підтримкою ECC).	Широкий діапазон DC (9–36V), підтримка зовнішніх UPS та резервування.	TPM 2.0, AES-NI (x86), підтримка TLS-offload на рівні мережевих чипів.	Розширений (−20…+70 °C).
Спеціалізований edge-модуль	Гетерогенна: ARM + NPU/GPU (AI-ядра). Апаратна віртуалізація (ARM-VE).	4–64 ГБ, Unified Memory (LPDDR5/LPDDR5x).	DC-джерело, інтегровані контролери Li-ion акумуляторів.	HSM, Secure Element (SE), ECC, TrustZone.	Індустріальний (−40…+85 °C).

Режими сну та пробудження (standby, suspend-to-RAM, глибокий сон із повним відключенням більшості доменів живлення) та циклічне керування живленням (duty-cycling) мережевих інтерфейсів (чергування інтервалів активного передавання з інтервалами вимкнення радіомодуля) у fog-вузлах є менш поширеними, ніж у сенсорних вузлах, але набувають актуальності для автономних та мобільних сценаріїв, де fog-вузол здатний періодично виконувати локальне оброблення і синхронізацію із хмарою. Для безпроводових інтерфейсів, зокрема Wi-Fi і стільникових модемів, duty-cycling використовують для зменшення споживання енергії за умови низької інтенсивності трафіку, що прямо впливає на доступний

бюджет обміну ключами, оновлення сертифікатів і виконання ресурсомістких криптографічних протоколів. В межах енергоорієнтованого планування завдань апаратні можливості енергоменеджменту визначають як керовані змінні, які є можливим інтегрувати до політик розподілу навантаження між локальним виконанням і перенесенням обчислень (offloading) у хмару, з урахуванням стану батареї та вимог щодо затримки.

Особливості експлуатації fog-вузлів у промислових, транспортних і зовнішніх умовах зумовлюють потребу врахування впливу температури, вібрацій, вологості, пилу та агресивних середовищ на електронні компоненти й підсистеми живлення. Температурні коливання, підвищена вологість і механічні вібрації здатні спричиняти деградацію параметрів DC-DC-перетворювачів та інших силових компонентів, що впливає на стабільність вихідної напруги, ефективність перетворення і, відповідно, на доступний енергетичний бюджет вузла [51]. Для промислових IoT-систем, які використовують безпроводові комунікації, вплив температури, вологості та вібрацій на характеристики радіомодулів та антен призводить до зниження якості каналу, підвищення ймовірності помилок і повторного передавання, що збільшує енергоспоживання на одиницю корисного трафіку.

Системи моніторингу стану fog-вузлів зазвичай містять сенсори температури ключових компонентів (стан заряду батареї, силові модулі, накопичувачі), датчики напруги й струму, а також засоби контролю швидкості обертання вентиляторів і стану елементів живлення, що дає змогу виявляти аномалії та запобігати аварійним відключенням. Інтеграція таких засобів моніторингу в систему керування енергоспоживанням дає змогу реалізувати політики, які в разі перевищення граничної температури або виявлення нестабільності електроживлення переходять до режимів зниженої продуктивності, часткового відключення функціональних модулів чи примусового завершення некритичних задач. Щодо інформаційної безпеки, такі реакції на фізичні фактори необхідно узгоджувати з політиками захисту: зниження частоти CPU чи вимкнення певних інтерфейсів у разі перегріву не має спричиняти некероване вимкнення криптографічних модулів або розрив критичних захищених

каналів; натомість є потреба в механізмах контрольованої деградації рівня захисту за збереження базового конфіденційного та цілісного режиму.

Зв’язок між механізмами апаратної адаптації та механізмами захисту інформації у fog-вузлах простежують насамперед через вплив вибору криптографічних алгоритмів, частоти обміну ключами та рівня протокольного захисту на навантаження CPU, використання пам’яті та, відповідно, на енергоспоживання й тепловиділення. Реалізація складних протоколів (TLS із сучасними параметрами ключів, DTLS, IPsec із частою ротацією ключів, механізми взаємної автентифікації та атестації) на ресурсно-обмежених вузлах без урахування енергетичних обмежень здатна зумовити обмеження доступних обчислювальних ресурсів для основних сервісів, збільшення затримок і в окремих випадках – відмову сервісів. Додаткове навантаження від криптографічного оброблення може також впливати на температуру SoC, що, у поєднанні з жорсткими умовами експлуатації, підвищує ризик перегріву і спрацьовування механізмів апаратного захисту (термальний тротлінг (thermal throttling), аварійне вимкнення).

Механізми адаптації електронного пристрою, такі як DVFS, режими енергозбереження, відключення модулів, є недоцільним аналізувати ізольовано від політик безпеки, оскільки агресивні енергозберігальні налаштування здатні обмежувати доступні криптопротоколи (через зниження продуктивності, зменшення доступної пам’яті, обмеження часу активності інтерфейсів), тоді як максимальний рівень захисту (наприклад, використання довгих ключів, часта ротація сесій, складні схеми автентифікації) стає несумісним із низькопотужним (low-power) режимом. Поєднання DVFS із адаптивними політиками планування здатне зменшувати енергоспоживання внаслідок балансування продуктивності та якості сервісу, що є можливим перенести на задачу узгодження криптографічного навантаження з енергетичними обмеженнями [52]. Для fog-вузлів це означає, що підсистема енергоменеджменту має містити інформацію про критичність захисних функцій і не може довільно відключати або деградувати компоненти, які відповідають за конфіденційність, цілісність і доступність даних, тоді як механізми захисту мають

враховувати поточний режим роботи пристрою та доступний енергетичний ресурс під час вибору криптопрофілів.

Формалізація узгодженого енергоорієнтованого керування ґрунтується на явному введенні залежності між енергетичним станом пристрою, режимом його апаратної конфігурації та рівнем реалізованих сервісів безпеки. У найпростішому якісному вигляді цю залежність можна подати функцією енергетичної вартості реалізації цільового рівня захисту, яка залежить від стану батареї, завантаження CPU та обраного криптопрофілю:

$$E = f(B, C, P) \tag{1.1}$$

де E – енергетичні витрати на реалізацію механізмів захисту за певний інтервал часу, B – показник стану заряду батареї, C – коефіцієнт завантаження процесора, P – вектор параметрів криптографічного профілю (тип алгоритмів, довжина ключів, частота ротації, параметри протоколів). Узгоджене керування вимагає, щоб у разі зміни B або C система могла адаптивно змінювати P, що забезпечить компроміс між цільовими показниками конфіденційності, цілісності та доступності та енергетичними обмеженнями пристрою.

Потреба в енергоорієнтованих механізмах захисту для fog-вузлів підтверджена аналізом досліджень із безпеки fog/edge-інфраструктур і апаратної безпеки IoT-кінцевих пристроїв, де підкреслено, що fog-вузли є ресурсно-обмеженими електронними пристроями, які одночасно мають забезпечувати оброблення значних обсягів трафіку, локальну аналітику та виконання функцій безпеки. Основні підходи до побудови політик безпеки для хмарних і традиційних IT-систем, які не враховують стану апаратури, енергетичних режимів та особливостей експлуатації fog-вузлів, спричиняють неефективне використання ресурсів, збільшення затримок, а також ризики відмов під час пікових навантажень або в умовах енергодефіциту [53]. Водночас дослідження енергоефективності IoT- та edge-пристроїв переважно спрямовані на раціоналізацію прикладних обчислень та передавання сигналів, тоді як інтеграцію енергоменеджменту з політиками безпеки згадують лише фрагментарно та

часто обмежують вибором полегшених алгоритмів без чіткої адаптації до поточного стану пристрою [54].

Аналіз публікацій за 2020–2024 роки виявляє обмежену кількість рішень, у яких здійснено тісну інтеграцію механізмів енергоменеджменту fog-платформ (DVFS, керування живленням модулів, моніторинг температури та стану батареї) з адаптивними політиками криптографічного захисту, що враховують як стан енергетичних ресурсів, так і динаміку загроз. Гетерогенність апаратури fog-рівня суттєво ускладнює узгодження механізмів безпеки, проте усталена модель, яка б поєднувала енергоорієнтовані механізми керування з адаптивним вибором криптопрофілю для забезпечення базових вимог конфіденційності, цілісності та доступності з уникненням неконтрольованих відмов, наразі відсутня [55]. Для fog-вузлів, які працюють у змінних умовах температури, вібрацій та доступності живлення, відсутні широко прийняті підходи до реалізації плавного зниження рівня захисту, коли у разі критично низького заряду батареї або перегріву пристрою відбувається контрольований перехід до зниженої, але не нульової якості захисту та сервісів. Загальна схема взаємодії описаних апаратних механізмів наведена на рис. 1.3.

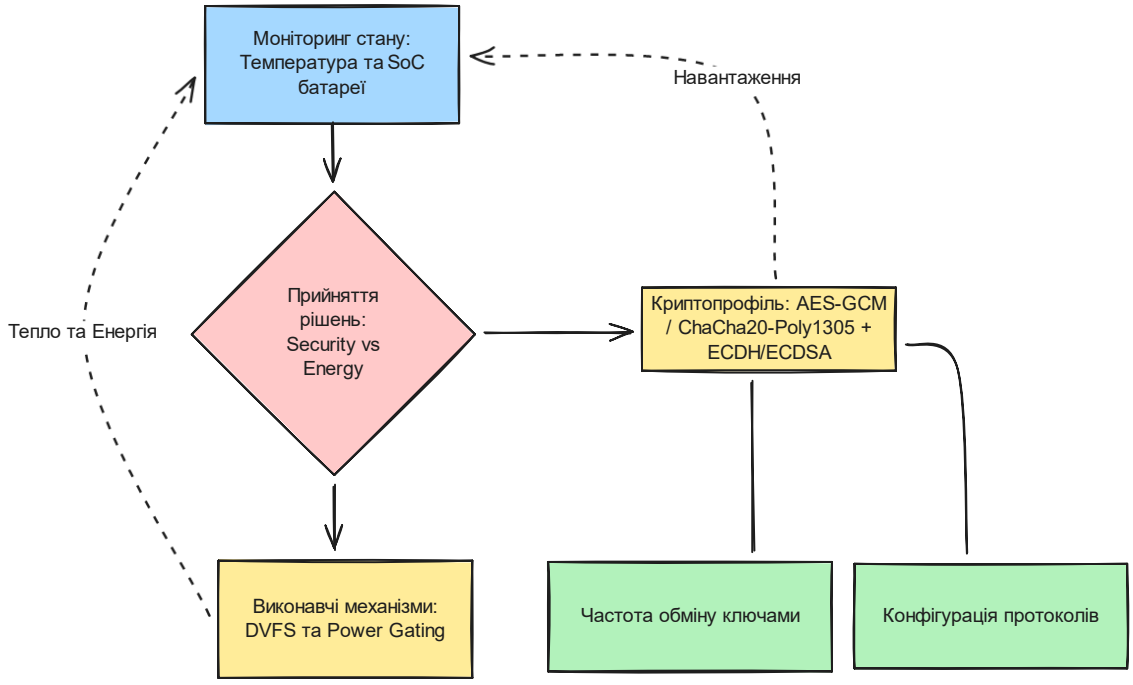

Рисунок 1.3 – Узагальнена схема взаємодії апаратних механізмів адаптації

1.5 Вимоги до механізму енергоорієнтованого криптозахисту вузла туманних обчислень

Сучасні туманні (fog) вузли, які побудовані на базі одноплатних комп’ютерів із застосуванням мікропроцесорних архітектур (наприклад, ARM SoC), мають обмежену обчислювальну потужність, лімітований енергетичний бюджет і жорсткі вимоги щодо теплорозсіювання порівняно з класичними серверними платформами. Результати експериментальних досліджень підтверджують суттєве зростання споживаної потужності й температури напівпровідникового кристала під час виконання ресурсоємних криптографічних задач [56]. У мобільних та автономних сценаріях функціонування (польові вимірювання, транспортні засоби, безпілотні апарати) fog-вузол переважно має електроживлення від акумуляторних батарей, що визначає енергоспоживання як критичний фактор безперервності його роботи. Специфіка збору даних у мобільному IoT поєднує високочастотні потоки телеметрії з процесами естафетного передавання (handover) та переривчастим зв’язком. Відповідно, fog-вузол виступає не лише вузлом агрегації шифрованих потоків, але й електронним пристроєм, енергетичний баланс якого визначає працездатність усієї підсистеми локального оброблення даних. З огляду на це, до механізму енергоорієнтованого захисту висувають низку специфічних апаратно-програмних вимог.

Енергоживучість fog-вузла визначають як здатність апаратно-програмного комплексу підтримувати працездатність ключових сервісів та базових механізмів захисту інформації за умов автономного електроживлення яке деградує (зокрема, під час автономної роботи в разі критичного зниження напруги акумулятора). Повна відмова локальних механізмів захисту внаслідок виснаження джерела електроживлення зумовлює утворення неконтрольованих каналів доступу. Тому механізм захисту має реалізовувати стратегію контрольованої деградації – поступове та кероване зниження обчислювального навантаження, пов'язаного з криптографічними перетвореннями (наприклад, перехід від складних схем взаємної

автентифікації до полегшених), за умови зниження рівня заряду батареї. Динаміку енергетичного стану вузла $E(t)$ у часі доцільно описати диференціальним рівнянням:

$$\frac{dE(t)}{dt} = -\left(P_{sys}(t) + P_{sec}(t)\right) \tag{1.2}$$

де $P_{sys}(t)$ – споживана потужність базового функціоналу електронної платформи (функціонування процесора, оперативної пам'яті, радіочастотних трансиверів), $P_{sec}(t)$ – додаткове енергоспоживання, зумовлене виконанням криптографічних примітивів та підтримкою протоколів захисту. Відповідно, впливи керування необхідно спрямовувати на мінімізацію $P_{sec}(t)$ у періоди енергетичного дефіциту за умови збереження мінімально необхідного рівня конфіденційності.

Криптографічні алгоритми мають різну енергетичну «вартість», яка залежить від довжини ключа, типу математичних операцій та наявності апаратних прискорювачів (наприклад, векторних інструкцій ARM NEON або спеціалізованих криптографічних співпроцесорів). Для узгодження вимог інформаційної безпеки з фізичними обмеженнями електронної апаратури механізм має функціонувати на основі дискретної множини криптографічних профілів (наприклад, енергоощадний, збалансований, максимальний). Кожен профіль p_i є можливим формалізувати у вигляді вектора параметрів $p_i = \{E_{hw}, L_{hw}, S_{level}\}$, де E_{hw} – апаратні енергетичні витрати (інтеграл споживаного струму), L_{hw} – апаратна латентність оброблення, S_{level} – розрахунковий показник рівня безпеки. Формулювання багатокритеріальної задачі вибору та безпосередній синтез інтегрального критерію якості для вибору профілю становлять предмет досліджень подальшого розділу дисертаційної роботи.

З позиції схемотехніки та конструювання електронної апаратури, високочастотні хаотичні перемикання режимів енергоспоживання процесора (наприклад, за технологією DVFS) або криптографічних профілів є недопустимими. Вони спричиняють імпульсні сплески споживаного струму, підвищують динамічне навантаження на інтегральні контролери живлення та генерують локальні теплові піки на кристалі. Для нівелювання цього ефекту необхідно застосовувати алгоритми з

гістерезисом під час оброблення сенсорних даних. Формалізовано, для показника рівня заряду батареї встановлюють порогові значення Th_{low} (нижній поріг) та Th_{high} (верхній поріг) для ухвалення рішення про зміну профілю.

Застосування такого підходу в комплексі з апаратним (RC-фільтри) та програмним (цифрові фільтри ковзного середнього) згладжуванням телеметричних показників дає змогу зменшити дисперсію енерговитрат і підвищити надійність функціонування вузла.

Ефективність адаптивного захисту безпосередньо залежить від підсистеми апаратного моніторингу. З боку платформи є вимога забезпечення безперервного збору телеметрії: напруги та струму розряду акумулятора (із застосуванням вбудованих аналого-цифрових перетворювачів (АЦП) мікроконтролера або спеціалізованих зовнішніх шунтів), температури напівпровідникових компонентів та рівня завантаження процесорних ядер. Агрегований показник рівня загроз можна обчислити не лише на основі реєстрації мережевих аномалій, але й з урахуванням апаратних маркерів. Зокрема, нетипові високочастотні флуктуації струму споживання або різке підвищення температури процесора за умов штатного мережевого навантаження можна ідентифікувати як ознаки реалізації атак через сторонні канали (Side-Channel Attacks) або спроб криптоаналітичного перебору, що слугує тригером для реконфігурації профілю захисту.

Механізм адаптації має вирізнятися детермінованою поведінкою: за ідентичних апаратних умов (температура, рівень заряду) та зовнішніх впливів (профіль загроз) система має демонструвати відтворюваний енергетичний відбиток. Це є критичною вимогою для проведення метрологічної оцінки, сертифікації IoT-устаткування та забезпечення модульності – відокремлення високорівневих політик керування безпекою від низькорівневої апаратної реалізації криптографічних примітивів. Узагальнення специфічних апаратно-програмних вимог до механізму енергоорієнтованого криптозахисту, які формують концептуальне підґрунтя для подальшого синтезу математичних моделей адаптивного керування, наведено в табл. 1.5.

Таблиця 1.5 – Узагальнені апаратно-програмні вимоги до механізму енергоорієнтованого криптозахисту fog-вузла

Вимога	Критерій виконання (Опис)	Параметри апаратної моделі	Методи апаратної реалізації
1. Енергоживучість	Здатність вузла виконувати критичні криптографічні функції за умов низького рівня заряду або живлення.	Залишковий заряд Q_{rem}, питома енергія на біт E_{bit}, час автономної роботи T_{life}.	Апаратне перемикання в режими глибокого сну (Deep Sleep), динамічне масштабування напруги та частоти (DVFS) на рівні мікроконтролера.
2. Керовані профілі	Наявність апаратних або програмно-апаратних пресетів захисту залежно від доступної енергії.	Індекс профілю P_{rid}, обчислювальна складність алгоритму $O(n)$, тактова частота f_{CPU} .	Використання алгоритмів полегшеної криптографії (Lightweight Cryptography), застосування апаратних прискорювачів на базі ПЛІС або крипто-співпроцесорів.
3. Стабільність перемикань	Відсутність критичних сплесків струму та апаратних збоїв під час динамічної зміни режимів захисту.	Перехідна затримка Δt_{sw}, амплітуда пікового струму I_{peak}, джитер.	Апаратне згладжування живлення (фільтрація), використання спеціалізованих контролерів живлення для плавного переходу між профілями.
4. Відтворюваність та стійкість	Гарантована ідентичність енергетичного відбитку для захисту від атак через сторонні канали (Side-Channel Analysis).	Дисперсія енерговитрат σ_E^2, кореляція енергетичних трас.	Апаратне маскування (Hardware Masking), внесення випадкових апаратних затримок на рівні логічних вентилів.
5. Критерії оцінки (Evaluation Criteria)	Наявність метрик ефективності для оцінки компромісу між рівнем безпеки та витратами енергії.	Коефіцієнт SER (Security-to-Energy Ratio), пропускна здатність на Ватт (Throughput per Watt).	Апаратне профілювання на стендах (Hardware-in-the-loop) із застосуванням прецизійних вимірювачів потужності та USB-тестерів.
6. Вимоги до моніторингу (Monitoring)	Безперервний зворотний зв'язок про стан живлення та використання апаратних ресурсів у реальному часі.	Частота дискретизації АЦП f_s, похибка вимірювання струму ΔI.	Використання вбудованих АЦП мікроконтролерів (наприклад, архітектури ESP32/ARM Cortex) або зовнішніх шунтів струму.
7. Адаптація до рівня загроз (Threat Adaptation)	Здатність системи змінювати параметри криптографічних примітивів відповідно до поточного рівня ризику в мережі IoT.	Ймовірність успішної атаки P_{atk}, ентропія апаратного джерела (TRNG).	Динамічна зміна довжини ключа, активація/деактивація апаратного генератора випадкових чисел (TRNG) для економії енергії.

Висновки до розділу

У розділі проведено комплексний аналіз архітектурних особливостей сучасних систем Інтернету речей та визначено критичну роль туманних (fog) вузлів у моделі Device–Fog–Cloud. Встановлено, що fog-вузли виконують функції локальної агрегації даних, попереднього оброблення і термінування захищених з'єднань та виступають критичними точками довіри в інфраструктурі. Водночас такі пристрої функціонують в умовах жорстких апаратних, енергетичних та теплових обмежень. Це створює фундаментальний конфлікт між вимогами щодо забезпечення високого рівня конфіденційності та цілісності даних, мінімізації мережевих затримок і збереження енергоресурсів пристрою.

З'ясовано, що наявні підходи до побудови політик безпеки для fog-рівня переважно використовують статичні або слабо адаптивні конфігурації захисту. Аналіз моделей збору даних та комунікаційних протоколів засвідчив, що застосування жорстко зафіксованих криптографічних профілів із максимальним рівнем захисту в умовах подієвого трафіку або критичного зниження заряду батареї спричиняє неефективне використання енергії, стрімке виснаження акумулятора та ризик повної відмови вузла.

Цю проблематику додатково ускладнюють сценарії мобільного IoT, де просторово-часова мобільність вузлів, нестабільність радіоканалів та процеси естафетного передавання (handover) зумовлюють часті розриви з'єднань. Потреба повторних процедур автентифікації генерує непропорційно високе навантаження на обчислювальні ресурси. Доведено, що контекстна адаптивність безпеки є системною вимогою для таких середовищ, де вибір криптопрофілю необхідно узгодити з поточним станом каналу, завантаженням процесора та залишковим зарядом батареї.

Аналіз публікацій у виданнях, індексованих у Scopus та Web of Science за 2022–2025 роки, засвідчує сформований дослідницький напрям. Дослідники успішно реалізують алгоритми типу ChaCha20-Poly1305 на вбудованих платформах ARM, підтверджуючи їхні переваги над AES за умов програмного виконання [57]. Пропоновані адаптивні схеми застосовують контекстне керування для контролю

доступу у Fog-середовищі [58]. Перспективним є використання нечіткої логіки для маршрутизації з урахуванням залишкового заряду [59]. Також існують динамічні схеми автентифікації для IoT-Fog середовищ [60]. Узагальнений аналіз виявляє ключову прогалину: у жодному з розглянутих досліджень не наведено єдиного механізму, що у режимі реального часу здійснює зміну симетричного алгоритму шифрування на основі безперервного моніторингу стану батареї.

На основі аналізу наявних прогалин сформульовано вимоги до механізму енергоорієнтованого криптографічного захисту fog-вузлів, базовим принципом якого визначено концепцію керованої деградації. Встановлено, що за критично низького рівня енергії або перегріву fog-вузол не має переходити у стан повної відмови, а має здійснювати контрольоване зниження рівня захисту зі збереженням його базових властивостей. Такий механізм вимагає наявності дискретної множини криптографічних профілів, здатних до динамічної адаптації до поточного профілю загроз (Threat Level), теплового стану пристрою та показників стабільності електроживлення.

Сукупність визначених вимог формує концептуальне підґрунтя для подання fog-вузла як об’єкта керування з відповідними параметрами стану (навантаження, заряд батареї, профіль загроз). З огляду на це виникає науково-практична задача формалізації математичної моделі «вартості» криптографічного захисту та синтезу інтегрального критерію якості J. Побудова системи рівнянь та функціоналів, які пов’язують енергетичні, часові та безпекові характеристики з параметрами керування, становитиме предмет досліджень у подальшому розділі дисертаційної роботи.

2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ

2.1 Модель вузла туманних обчислень як об’єкта керування

Туманний (fog) вузол у системі IoT–fog–cloud доцільно аналізувати як керований об’єкт із енергетично та обчислювально обмеженими ресурсами, на який діють як корисні сервісні навантаження, так і потенційно ворожі впливи трафіку. Для мобільних та енергообмежених fog-вузлів, зокрема на базі одноплатних ARM-комп’ютерів Raspberry Pi, політики енергокерування та конфігурація механізмів інформаційної безпеки істотно впливають на затримку, пропускну здатність і загальну якість сервісу в Fog-IoT системах [61].

У термінах теорії автоматичного керування fog-вузол описано вектором стану $\bar{x}_k$, на який впливають керувальні дії $\bar{u}_k$ та збурювальні впливи $\bar{w}_k$, а також задають вимірювані вихідні змінні y_k, що характеризують якість функціонування. Дискретний час k відповідає періодам прийняття рішень контролером безпеки та енергокерування, які синхронізовані з періодом збору телеметрії й оновлення показників загроз. У загальному вигляді динаміку fog-вузла як об’єкта керування подають у формі різницевої моделі:

$$x_{k+1} = f(\bar{x}_k, \bar{u}_k, \bar{w}_k), \quad y_k = g(\bar{x}_k, \bar{u}_k), \tag{2.1}$$

де функція f описує еволюцію стану енергоресурсів і навантаження, а g – вихідні показники якості сервісу та безпеки. Надалі саме вибір структури вектора стану $\bar{x}_k$ та керування $\bar{u}_k$ визначає можливості побудови адаптивного механізму вибору криптографічного профілю.

Структурну схему fog-вузла як об’єкта керування подано на рис. 2.1.

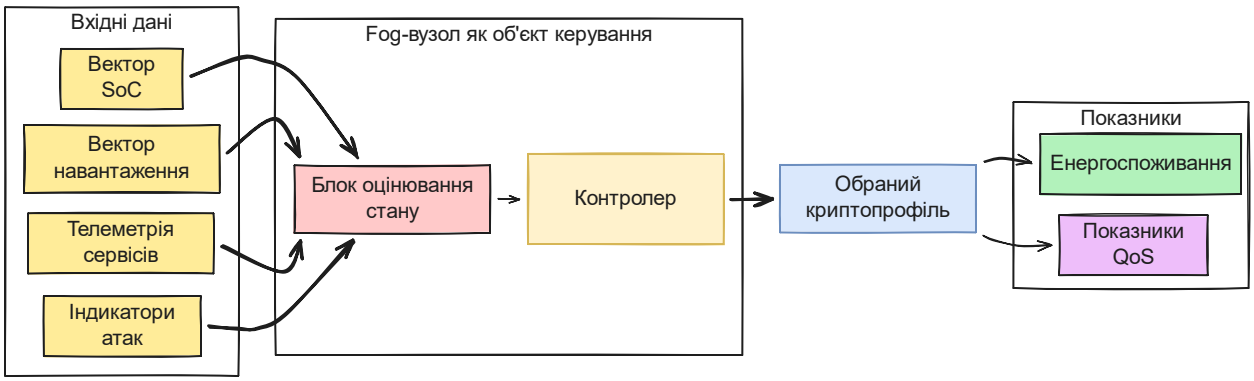

Рисунок 2.1 – Узагальнена структурна схема fog-вузла як об’єкта керування

Вектор вхідних сигналів (або компонентів стану) моделі $\bar{x}_k$ поєднує енергетичні, обчислювальні, сервісні та безпекові параметри, які формують як результат вимірювань локальних сенсорів fog-вузла та модулів моніторингу мережевого трафіку. До ключових компонентів x_k належить стан заряду акумулятора, що в теорії енергетичних систем визначають як відношення поточного доступного заряду до номінальної ємності батареї й використовують як головний індикатор залишкового енергоресурсу. Для мобільних fog-вузлів рівень заряду батареї обмежено діапазоном $[0;1]$ або у відсотковому діапазоні $0\ldots100\%$, а політики керування зазвичай оперують пороговими зонами (наприклад, критична зона нижче певного рівня заряду, робоча зона та зона резерву). У роботах із дискретного керування зарядом та оцінювання стану заряду батарей у вбудованих SoC-системах модель батареї, як правило, задають у дискретно-часовій формі з кроком у частки або одиниці секунд, що узгоджено з періодом зняття телеметрії та оновлення рішень енергоконтролера [62].

Точність оцінки рівня заряду батареї визначає допустимий рівень енергоспоживання fog-вузла і, відповідно, множину криптографічних і протокольних механізмів, які можливо активувати без загрози передчасного виснаження батареї. У дослідженнях з енергокерування в IoT-системах заряд батареї часто розглядають як ключовий стан або обмеження, що впливає на вибір рівня захисту та політик енергоспоживання, а отже – на досяжний час автономної роботи пристрою [63]. Для fog-вузлів на базі Raspberry Pi, які здатні працювати як від мережі, так і від

автономних джерел, така постановка зумовлює потребу динамічно пов’язувати політики безпеки з оцінкою залишкового енергоресурсу.

Навантаження fog-вузла описано сукупністю показників завантаження CPU, обсягу оперативної пам’яті, інтенсивності вхідного та вихідного трафіку, кількості активних сесій і потоків, а також використання апаратних прискорювачів криптоопераций за їхньої наявності. Збільшення завантаження CPU внаслідок паралельного виконання сервісних задач і криптографії спричиняє помітне зростання споживаної потужності та температури, що безпосередньо впливає на тривалість автономної роботи. Інтенсивність трафіку, яку зазвичай описують статистиками пакетів за одиницю часу й числом активних сесій, визначає частоту виконання криптоопераций (шифрування, автентифікація, встановлення сеансів), що прямо впливає на середнє енергоспоживання та затримки. Fog-вузол є ресурсно-обмеженим сервером, у якому криптографічні задачі конкурують за CPU та пам’ять із застосунками оброблення даних, тому в умовах високого навантаження допустимий рівень складності криптопрофілю необхідно обмежувати адаптивними політиками керування.

Телеметрія сервісів, що функціонують на fog-вузлі, охоплює показники якості обслуговування (Quality of Service, (QoS)): затримку (latency) на оброблення й маршрутизацію пакетів, частку втрачених пакетів (packet loss), пропускну здатність (throughput), а також індикатори стану критичних застосунків (наприклад, сервісів керування в реальному часі або медичних IoT-сервісів). Параметри QoS є ключовими вихідними характеристиками, які визначають прийнятність того чи іншого рівня криптографічного захисту в умовах затримок, обмежених вимогами застосунку. Надмірно ресурсомісткі криптографічні профілі здатні зумовити зростання часу оброблення пакетів до неприйнятних значень у системах із жорсткими обмеженнями щодо latency, тому вибір профілю має завжди враховувати допустимі межі затримок і втрат для кожного класу сервісів.

Індикатори атаки формують на основі результатів систем виявлення вторгнень (Intrusion Detection Systems, (IDS)), аналізу аномалій у структурі та обсязі трафіку, частоти помилок автентифікації, а також кореляції подій безпеки на рівні fog-вузла.

Індикатори атак охоплюють частоту аномальних з’єднань, обсяги підозрілого трафіку, збільшення числа відмов автентифікації й сигнали від локальних модулів аналізу поведінки. Ці показники доцільно агрегувати у скалярний або векторний показник рівня загроз (Threat Level) T_k, який надалі використовують як один із ключових вхідних сигналів керування криптографічним профілем.Сукупність описаних вхідних параметрів моделі формує вектор:

$$\bar{x}_k = [\bar{B}_k, \bar{C}_k, \bar{N}_k, \bar{Q}_k, \bar{T}_k]^T, \tag{2.2}$$

де $\bar{B}_k$ – стан заряду батареї; $\bar{C}_k$ – завантаження процесора; $\bar{N}_k$ – показники інтенсивності трафіку та кількості сесій; $\bar{Q}_k$ – вектор показників якості сервісу; $\bar{T}_k$ – агрегований індикатор рівня загрози. Такий вектор є входом до адаптивного контролера політик безпеки, який синтезує рішення щодо конфігурації механізмів захисту залежно від поточного стану системи. Приклади вхідних параметрів моделі наведено у табл. 2.1.

Таблиця 2.1 – Приклади вхідних параметрів моделі

Параметр	Одиниця вимірювання	Діапазон значень	Періодичність вимірювання	Тип джерела даних
SoC Temperature (Температура кристала)	°C	+30 ... +105	1–5 с	Апаратна телеметрія (BMS/термодатчики)
CPU Load (Завантаження процесора)	%	0 ... 100	500 мс – 2 с	Системні лічильники ОС (профайлер)
Latency (Затримка передавання)	мс	1 ... 500+	Кожен пакет / 1 с	IDS / Мережевий стек ОС
Packet Loss (Втрата пакетів)	%	0 ... 100	5–10 с (вікно спостереження)	IDS / Мережевий стек ОС
Throughput (Пропускна здатність)	Мбіт/с (Gbps)	Відповідно до ТТХ каналу	1 с	IDS / Системні лічильники ОС
Частота помилок автентифікації	спроб/хв	0 ... 60+	Подійна (Event-driven)	IDS / Логи безпеки ОС
Індекс аномальності трафіку	відн. од. (score)	0.0 ... 1.0	5–30 с	IDS (Апаратні/програмні модулі аналізу)

Керувальний вихід моделі u_k інтерпретують як вибір криптографічного профілю або режиму захисту fog-вузла, що визначає набір застосовуваних алгоритмів, довжини ключів, параметри протоколів та політики оновлення ключів. У межах підходу Security-by-Contract безпекову конфігурацію пристрою задають у вигляді формалізованого контракту (профілю поведінки), який обирають і застосовують на вузлі залежно від політик та умов оточувального середовища [64]. Для енергообмежених fog-вузлів доцільно ввести скінченну множину припустимих криптографічних профілів U, де кожному профілю відповідає фіксований набір алгоритмів шифрування, цілісності, автентифікації та параметрів протоколів (наприклад, рівень TLS/DTLS, параметри OSCORE, режими MQTT-TLS).

Дискретна керувальна змінна тоді описана як:

$$\bar{u}_k \in U = \{P_{Safe}, P_{Balanced}, P_{Max}, P_{Survival}\}, \tag{2.3}$$

де профілі P інтерпретують як «Safe», «Balanced», «Max», «Survival» із відповідними параметрами. Для порівняння криптографічних алгоритмів на вбудованих платформах доцільно задавати для кожної реалізації нормовані показники обчислювальної та енергетичної вартості (наприклад, на одну криптографічну операцію або на одиницю оброблених даних), а також оцінювати вплив на часові характеристики оброблення мережевого трафіку, зокрема на затримку шифрування пакетів. Для IoT–fog-сценаріїв такі профілі доповнюють оцінками рівня безпеки, які базують на рекомендаціях стандартів і криптоаналітичних результатах, що створює основу для подальшого формального опису компромісу «енергія–затримка–безпека».

Різні режими функціонування fog-вузла суттєво впливають на допустимий простір керувальних дій, тобто множину профілів, які є можливим обрати в певний момент часу. Для опису функціональних режимів введено дискретну змінну режиму M_k, яка набуває значень із множини $M_k \in \{M_{normal}, M_{threat}, M_{survival}\}$, де M_{normal} відповідає штатній роботі без ознак атак і з достатнім енергоресурсом, M_{threat} – режиму підвищеного рівня загроз, $M_{survival}$ – режиму енергетичної деградації. У контексті енергобезпечних протоколів для IoT доцільно використовувати спеціальні

режими деградації, за яких в умовах критично низького рівня заряду акумулятора частину несуттєвих сервісів вимикають, а механізми захисту переводять у спрощену конфігурацію, що забезпечує мінімально допустимий рівень безпеки за умови збереження живучості системи. Діаграма режимів fog-вузла наведена на рис. 2.2.

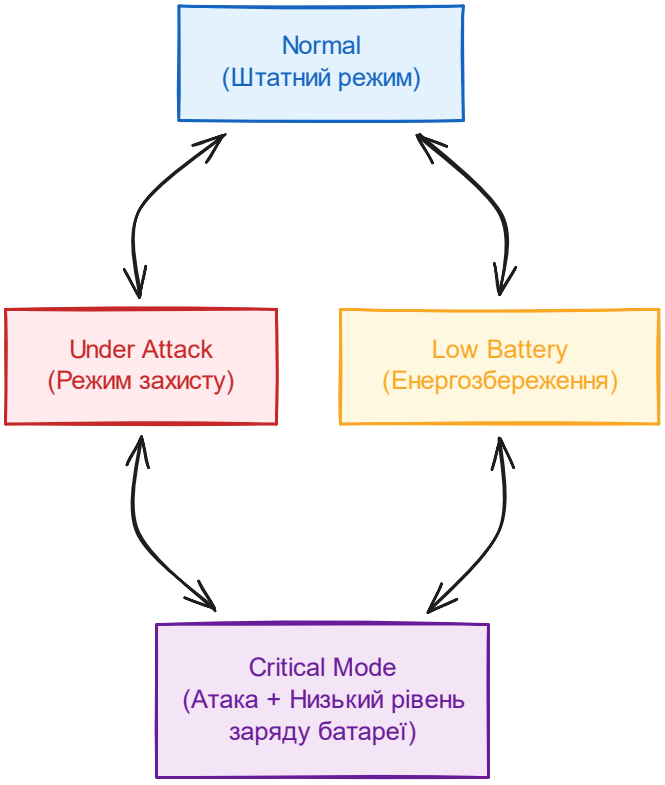

Рисунок 2.2 – Діаграма режимів fog-вузла

Функціонування fog-вузла здатне відбуватися в комбінованих режимах, наприклад «атака за низького заряду батареї», що описано поєднанням індикатора високого рівня загроз M_{threat} та належності рівня заряду батареї до критичної зони. Такі сценарії створюють конфлікт між потребою підвищити рівень захисту через активну атаку та необхідністю зберегти залишковий ресурс батареї, що безпосередньо впливає на стратегію вибору криптопрофілю. Перехід між режимами M_k залежить від порогів для SoC, індикаторів атак, а також політик оператора. Політики переходів мають враховувати як технічні, так і організаційні вимоги безпеки.

Залежність між режимами роботи fog-вузла та керувальними впливами відображено через обмеження на допустиму множину профілів у кожному режимі. Для режиму M_{normal} припущено повний простір U або його суттєву частину, що дає

змогу використовувати криптопрофілі з підвищеними вимогами щодо енергоспоживання та затримок за умови наявності достатнього енергоресурсу й резервів продуктивності. Для режиму $M_{survival}$ визначають підмножину $U_{survival} \subset U$, до якої входять енергоощадні профілі, можливо зниженої криптостійкості або спрощених протоколів, що забезпечують виконання базових функцій системи за мінімального споживання енергії. У режимі M_{threat} множину допустимих профілів U_{threat} зміщують до профілів із вищим рівнем захисту, однак включаючи обмеження, які накладено поточним значенням рівня заряду батареї та навантаженням.

За таких умов задачу керування криптографічним профілем формально формулюють як знаходження відображення:

$$\bar{u}_k = \pi(\bar{x}_k, M_k), \tag{2.4}$$

де π – політика або закон керування, що відображає вектор вхідних сигналів $\bar{x}_k$ та режим M_k у дискретне рішення щодо профілю. У літературі описано підходи, де π реалізують у вигляді систем правил, нечіткого керування або політик, заснованих на машинному навчанні, які динамічно адаптують параметри безпеки залежно від навантаження та загроз. Для енергообмежених fog-вузлів ці механізми мають залишатися обчислювально легкими, працювати в реальному часі на ARM-платформах та не спричиняти істотного зростання загального енергоспоживання вузла [65].

Вектор стану fog-вузла $\bar{x}_k$ доцільно вибудовувати так, щоб він відображав як довгострокові енергетичні характеристики, так і короткочасну динаміку навантаження й безпекової ситуації. До компонентів $\bar{x}_k$ раціонально додати поточний активний криптографічний профіль, згладжені показники навантаження та заряду батареї, а також оцінку рівня загрози. Отже, розширений вектор стану можна подати у вигляді:

$$\bar{x}_k = \{\bar{u}_{k-1}, B_k, C_k, Q_k, T_k\}, \tag{2.5}$$

де $\bar{u}_{k-1}$ – активний криптографічний профіль на попередньому кроці, B_k і C_k – згладжені показники енергії та навантаження, Q_k – агреговані метрики QoS, T_k – агрегований індикатор загрози. Для адаптивного керування ресурсами у fog-середовищі застосовують дискретну часову модель із фіксованим кроком оновлення стану. Інтервал у декілька десятків секунд забезпечує достатню чутливість до змін навантаження та стану безпеки, водночас не вимагає постійного перерахунку політики й зберігає низькі обчислювальні накладні витрати.

Вибір кроку дискретизації Δt, який пов'язує фізичний час t із дискретним індексом k, визначають компромісом між швидкістю реакції контролера й стабільністю прийнятих рішень. Занадто малий крок спричиняє високі накладні витрати на обчислення й потенційну нестабільну поведінку системи через часті перемикання профілів, тоді як надто великий крок здатний зробити систему інертною до короткочасних піків навантаження або раптових атак. У роботах з аналізом оркестрації сервісів у fog-інфраструктурах, рекомендації щодо вибору періоду моніторингу й оновлення політик прив'язані до характеристик застосунків, їхніх вимог до затримки та частоти оновлення телеметрії, що підкреслює доцільність параметричного налаштування часової дискретизації під конкретні сценарії використання [66]. Приклади вибору кроку дискретизації Δt наведено у табл. 2.2.

Таблиця 2.2 – Приклади вибору кроку дискретизації Δt для різних класів застосунків

Клас застосунків	Типові вимоги до затримок	Характер динаміки навантаження	Рекомендований діапазон Δt
Промислові IoT-системи (IIoT)	Низькі та детерміновані (10–50 мс)	Висока циклічність, критичність до джиттера, потік телеметрії від сенсорів.	1 с – 5 с
Медичні сервіси (IoMT)	Середні (100–500 мс) для моніторингу; критичні для екстрених подій.	Стабільний потік даних із періодичними сплесками під час виявлення аномалій (аритмія тощо).	2 с – 10 с
Інтелектуальний транспорт (ITS)	Дуже низькі (<20 мс) для безпеки руху; вищі для інфотейнменту.	Висока динамічність через мобільність вузлів, різка зміна топології та щільності запитів.	0.5 с – 2 с
Розумне місто (Smart City)	Високі до середніх (1–5 с)	Повільні зміни (трафік, освітлення), великі масиви агрегованих даних.	10 с – 60 с

Формулювання задачі керування на концептуальному рівні базують на потребі одночасного забезпечення вимог до конфіденційності, цілісності й доступності (CIA – Confidentiality, Integrity, Availability) оброблюваних даних, енергоживучості fog-вузла та прийнятних показників QoS. У контексті IoT та IIoT-систем із використанням fog-обчислень конфігурація механізмів захисту має враховувати обмежений енергетичний бюджет вузлів. Застосування надмірно ресурсомістких засобів безпеки на пристроях із живленням від батареї зумовлює підвищене енергоспоживання, скорочення часу автономної роботи та, як наслідок, погіршення загальної надійності та доступності системи. Для мобільних fog-вузлів задача керування ускладнена тим, що зміни енергетичного стану та рівнів загрози відбуваються на різних часових шкалах, що вимагає багаторівневого підходу до синтезу політики.

У загальному вигляді задачу вибору криптопрофілю формулюють як задачу багатокритеріального вибору, метою якої є мінімізація інтегрального критерію якості із забезпеченням мінімально допустимих рівнів безпеки та якості сервісу. Для цього вводять функціонал якості (інтегральний критерій) J, що агрегує складові енергоспоживання, затримок та ризику безпеки за період спостереження:

$$J = \sum_{k=0}^{N} L(\bar{x}_k, \bar{u}_k)\,, \tag{2.6}$$

де $L(\bar{x}_k, \bar{u}_k)$ – миттєва функція якості, яка зростає в разі зменшення рівня заряду батареї, погіршення QoS або зниження рівня захисту в умовах підвищеного рівня загрози. Аналогічні функціонали є ефективними для балансування між споживаними ресурсами та рівнем захисту під час розгортання IDS, фільтрації трафіку й криптографічних сервісів на fog-вузлах.

У межах цієї роботи конфігурацію безпеки IoT-пристроїв задають у формі контрактів, які описують припустиму поведінку та очікувані механізми захисту. Fog-вузли розглядають як довірені посередники, що здійснюють перевірку контрактів на відповідність політикам мережі та ресурсним можливостям. Процес вибору та активації профілю безпеки інтерпретують як задачу задоволення обмежень, у якій, поряд із формальними правилами політики, враховано енергетичні обмеження fog-

вузла, вимоги до QoS, а також динамічні показники, такі як заряд батареї, рівень навантаження й поточний стан загроз.

У наявних оглядах із безпеки fog-систем, включно з підходами на основі машинного навчання, енергетичну складову переважно згадують як обмеження ресурсів вузлів, тоді як формалізовані математичні моделі комбінованого енергобезпекового компромісу для fog-вузлів (з явним урахуванням стану заряду батареї та дискретних профілів безпеки як керувальних дій) практично не описані [67].

Сформульована математична модель fog-вузла як динамічного об’єкта керування з вектором стану $\bar{\boldsymbol{x}}_k$, вхідними сигналами $\bar{w}_k$, режимами M_k та дискретним керуванням $\bar{u}_k$ створює основу для побудови подальших компонентів енергобезпекової постановки задачі. Для отримання повноцінної моделі є необхідність доповнити опис відкритою моделлю загроз, яка визначає типові вектори атак, їхню інтенсивність і вплив на індикатори T_k, а також моделлю якості криптографічного захисту, що формалізує залежності енергоспоживання та затримок від обраного профілю на цільових ARM-платформах. Подальший синтез інтегрального критерію якості J, що поєднує енергетичні, часові та безпекові складові в єдиному функціоналі, дасть змогу визначити задачу керування як процедуру адаптивного вибору та обґрунтувати застосування адаптивного механізму вибору криптопрофілю на основі нечіткого (FLC) чи правилового контролера.

2.2 Модель загроз та формування рівня загрози для вузла туманних обчислень

Вузол туманних обчислень (fog node) у трирівневій архітектурі аналізують як проміжний апаратний рівень оброблення даних, що акумулює телеметрію від кінцевих пристроїв та сервісні запити з боку хмари. Така концентрація інформаційних потоків перетворює мікропроцесорну підсистему fog-вузла на головну мішень для атак. Для побудови енергоорієнтованого механізму керування необхідним є введення формалізованого показника рівня загрози (Threat Level), який кількісно описує

поточний ступінь небезпеки з урахуванням інтенсивності апаратних переривань та сигналів систем виявлення вторгнень (Intrusion Detection Systems, (IDS)).

2.2.1 Класифікація загроз в аспекті впливу на електронні компоненти вузла

У середовищі Інтернету Речей туманні вузли піддаються комплексному впливу загроз, які доцільно класифікувати за ступенем їхнього навантаження на апаратні компоненти. Мережеві атаки (сканування портів, підроблення пакетів) спричиняють лавиноподібне зростання переривань від мережевого контролера, що виводить центральний процесор з режимів енергозбереження. Основні типи загроз та їхній вплив на електронну підсистему наведено в табл. 2.3.

Таблиця 2.3 – Основні типи загроз для fog-вузла

Тип загрози	Приклади атак	Телеметричні / IDS-індикатори	Очікуваний вплив на Threat Level
Мережеві (Network-based)	DDoS, Flooding (SYN/UDP), Man-in-the-Middle (MitM)	Packet drop rate, network throughput, inter-arrival time (IAT), кількість активних з'єднань	High / Critical (залежно від інтенсивності)
Навантажувальні (Resource Exhaustion)	CPU exhaustion, RAM bloating, Storage filling	CPU load, memory usage, queue length, температура чипа (thermal throttling)	Medium / High (призводить до відмови в обслуговуванні)
Протокольні (Protocol-specific)	Sinkhole, Sybil attack, Wormhole, маніпуляція MQTT/CoAP	Routing table instability, control packet overhead, IDS anomaly score, помилки автентифікації	High (порушує топологію мережі)
Конфіденційність та цілісність	Data tampering, Side-channel attacks, ін'єкція хибних даних	Hash mismatch, несанкціоновані API-виклики, раптові зміни ентропії даних, IDS signature match	Critical (пряма загроза даним користувача)

Протокольні загрози, такі як спроби навмисного ініціювання ресурсомістких процедур встановлення з'єднання (handshake), генерують максимальне навантаження на арифметико-логічний пристрій мікропроцесора під час виконання операцій з відкритим ключем. Для fog-вузлів, що підтримують одночасну роботу багатьох

клієнтів, такі атаки зумовлюють швидке вичерпання оперативної пам'яті (Random Access Memory, (RAM)) та переповнення апаратних буферів.

В архітектурах промислового Інтернету Речей (Industrial IoT, (IIoT)) fog-рівень є точкою конвергенції керуючих сигналів. Відповідно, компрометація електронних компонентів на цьому рівні має системний ефект для всієї інфраструктури, що робить цей вузол оптимальним місцем для розгортання механізмів адаптивного криптографічного профілювання.

2.2.2 Навантажувальні атаки як тригер виснаження автономного джерела живлення

Навантажувальні атаки (типу DDoS) на fog-вузли характеризують генерацією аномальних обсягів атакувального трафіку, що зумовлює деградацію якості обслуговування. В апаратному аспекті такі атаки розглядають як інструмент примусового утримання мікропроцесора у стані максимальної продуктивності. До типових індикаторів цього процесу належать аномальне зростання пікового струму споживання CPU, збільшення довжини черг команд та підвищення температури чипа. Агресивні хвилі запитів (burst-патерни) істотно змінюють профіль струмоспоживання вузла навіть за відсутності явного перевантаження оперативної пам'яті.

В енергообмежених конфігураціях навантажувальні атаки мають подвійний ефект: вони не лише збільшують затримки оброблення корисних сигналів, але й критично прискорюють розряд батареї внаслідок безперервної роботи радіотрансивера та підсистеми пам’яті. Зловмисник, навмисно перевантажуючи fog-рівень, здатний ініціювати небажаний процес делегування обчислень у хмару (offloading) внаслідок спрацювання захисних механізмів контролера живлення.

З позиції формування агрегованого показника загроз (Threat Level) зростання мережевого й обчислювального навантаження інтерпретують як пряму загрозу енергетичній живучості вузла. Як вхідні ознаки для виявлення атаки застосовують середні й дисперсійні значення завантаження CPU, інтенсивність апаратних

переривань, частоту помилок криптографічного модуля та швидкість падіння напруги на джерелі електроживлення.

Для fog-вузлів, що працюють у відриві від стаціонарної електромережі, атакувальні сплески спричиняють суттєве збільшення миттєвого енергоспоживання та різке скорочення часу автономної роботи. У таких умовах DDoS-атаки слід розглядати передусім як ресурсовиснажувальний вплив на батарею (Battery Exhaustion Attack). У контексті адаптивного керування це означає, що ідентифікація навантажувальної атаки має зумовлювати миттєве підвищення показника Threat Level, що, своєю чергою, ініціює перегляд криптографічного профілю для захисту залишкового заряду.

2.2.3 Формалізація агрегованого індикатора загроз

Для цілей керування криптографічним профілем fog-вузла доцільно вводити скалярну величину Threat Level T_k, яка відображає агреговану оцінку поточного рівня загроз у дискретні моменти часу k на основі багатовимірного вектора ознак F_k. Інтегральний ризиковий показник обчислюють з урахуванням параметрів контексту сесії, чутливості ресурсів, серйозності дії та історії поведінки або ризику для пристрою, і використовують для прийняття динамічних рішень щодо доступу. Такий показник відповідає спеціалізованій формі рівня загрози (Threat Level), що агрегує кілька контекстних факторів у єдину числову метрику [68].

У загальному вигляді Threat Level можна визначити як відображення:

$$T_k = \Phi(F_k, W), \tag{2.7}$$

де F_k містить мережеві метрики (обсяг і структуру трафіку, інтенсивність з’єднань, частоту помилок), телеметрію системи (завантаження CPU, затримки, довжини черг), показники IDS (оцінки аномальності, типи виявлених атак) та політики безпеки, а W позначає набір параметрів агрегування, вагових коефіцієнтів та порогів. Реалізацію функції Φ описують різними підходами: від лінійної або вагової

суми нормованих ознак до використання нечітких моделей, байєсівських оцінок та методів машинного навчання для отримання згладженого ризикового показника.

Властивість інерційності Threat Level досягають через внесення часової залежності та згладжувальних фільтрів у формулу оновлення. Один із підходів полягає у використанні експоненціального згладжування:

$$T_k = \alpha \cdot \Phi(F_k) + (1 - \alpha) \cdot T_{k-1}, \tag{2.8}$$

де $\Phi(F_k)$ – миттєва оцінка рівня загроз на основі поточного вектора ознак F_k, а $\alpha \in [0,1]$ – параметр згладжування, що задає компроміс між чутливістю до нових подій і стійкістю до одиничних сплесків. Нечітку модель оцінки ризику доступу у Fog-IoT проєктують так, щоб не реагувати на дрібні флуктуації телеметричних ознак: для цього вводять механізми згладжування та гістерезису, які запобігають частим перемиканням режимів безпеки й забезпечують стабільні рішення. Схематичну модель формування Threat Level наведено на рис. 2.3.

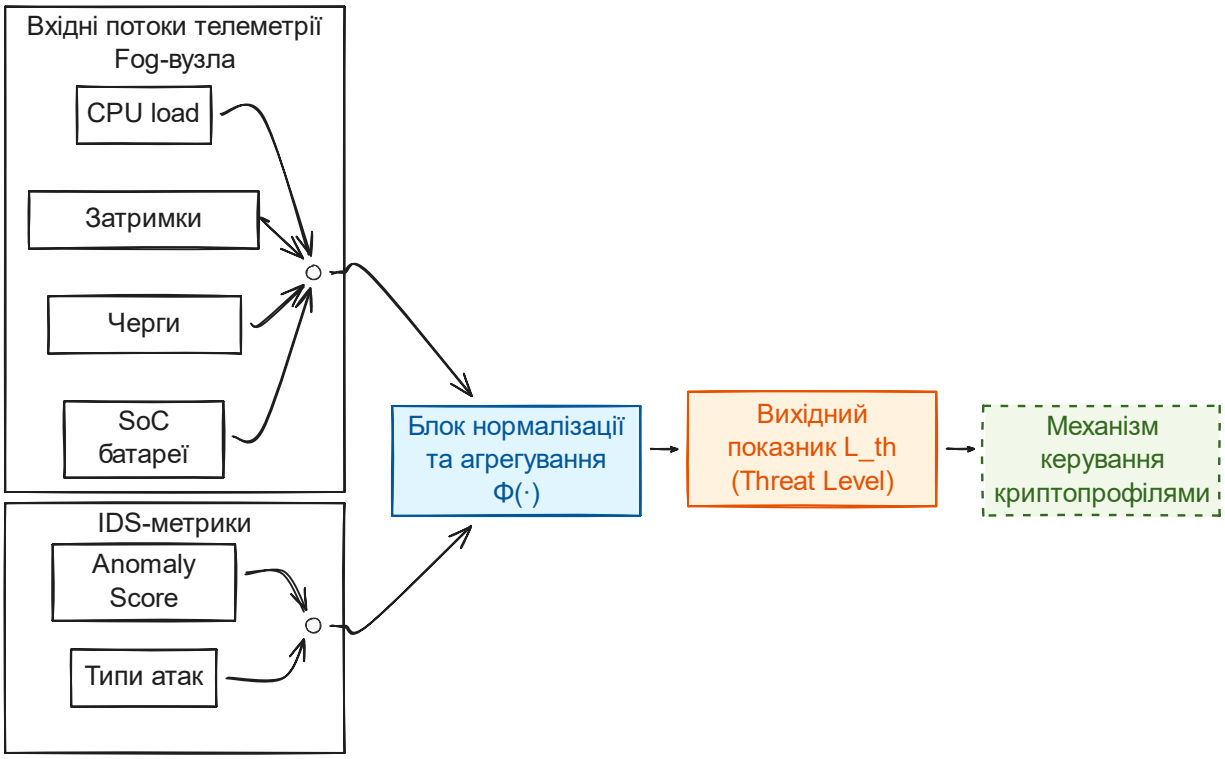

Рисунок 2.3 – Схематична модель формування Threat Level

Нормалізація Threat Level до інтервалу [0,1] або [0,100%] спрощує його використання як керувального сигналу в адаптивних алгоритмах та дає змогу вводити порогові значення для класифікації станів системи, наприклад, "низький", "середній", "високий" рівні загрози. Для fog-вузлів з енергообмеженими ресурсами доцільно включати в модель Threat Level не лише класичні мережеві та IDS-показники, але й додаткові енергетичні параметри (заряд батареї, оцінку залишкового часу роботи).

2.2.4 Джерела даних для формування рівня загрози

Джерела даних для оцінки Threat Level на fog-рівні поділяють на два основних класи: телеметрію fog-вузла (системні й прикладні показники, які збирають засобами моніторингу) та метрики систем виявлення вторгнень, розгорнутих безпосередньо на fog-рівні або у тісній взаємодії з ним. Телеметрія охоплює завантаження процесора, використання пам’яті, параметри мережевих інтерфейсів (швидкість прийому/передавання, кількість помилок, перезапусків, відкинутих пакетів), характеристики черг і затримок, а також прикладні показники, специфічні для сервісів fog-вузла. У QoS-орієнтованих моделях fog-комп’ютингу системну телеметрію вузлів використовують для динамічного балансування навантаження та вибору місця розміщення задач. У запропонованому підході ті самі показники додатково розглядають як вхідні параметри для формування інтегрального показника Threat Level fog-інфраструктури.

IDS-метрики охоплюють оцінки аномальності трафіку, ймовірності належності потоку до певного класу атак, позначки виявлених інцидентів, типи атак (DDoS, сканування, brute-force, ін’єкції), а також внутрішні показники довіри до джерел трафіку або пристроїв. Локальне розгортання моделей машинного навчання безпосередньо на fog-вузлах, зокрема глибоких архітектур (згорткових нейронних мереж (CNN), мереж довгої короткострокової пам'яті (LSTM), гібридних ансамблів), дає змогу виявляти широкий спектр загроз із малою затримкою та знижувати трафік до хмари. Водночас такі IDS мають проєктувати з урахуванням жорстких

обчислювальних і, відповідно, енергетичних обмежень fog-апаратури, що потребує застосування полегшених моделей та оптимізації їхньої складності [69].

У мінімальній конфігурації Threat Level формують виключно на основі телеметрії fog-вузла без використання повноцінної IDS, наприклад, шляхом виявлення різких змін використання CPU, збільшення затримок і довжини черг, появи піків помилок та нестандартних патернів навантаження. Такий підхід забезпечує базову чутливість до навантажувальних атак навіть в енергообмежених конфігураціях, де розгортання складних моделей IDS є небажаним або неможливим. За наявності розгорнутої IDS Threat Level доповнюють інформацією про класи атак та рівні довіри до окремих потоків, що дає змогу диференційовано реагувати на різні типи загроз (наприклад, посилювати конфіденційність під час конфіденційнісно-орієнтованих атак, зосереджуватися на доступності під час DDoS).

Модель Threat Level має залишатися працездатною за відсутності розгорнутої IDS, використовуючи більш грубі, але доступні показники, що відповідає вимогам до мінімального функціоналу захисту в енергообмежених середовищах. У таких сценаріях доцільно застосовувати прості евристичні або нечіткі правила, які на основі телеметрії fog-вузла визначають рівні станів "нормальна робота", "підвищене навантаження", "ймовірний напад", із подальшим відображенням цих станів на шкалу Threat Level.

2.2.5 Зв’язок рівня загрози із вимогами щодо конфіденційності, цілісності, доступності та живучості вузла туманних обчислень

Класична триада CIA (Confidentiality – конфіденційність, Integrity – цілісність, Availability – доступність) у контексті fog-вузлів доповнена вимогою щодо живучості (resilience), тобто здатності підтримувати прийнятний рівень сервісів та захисту в умовах атак і деградації ресурсів. Ризик-орієнтовані моделі для Fog-IoT пов’язують оцінку ризику (яка за змістом є аналогом Threat Level) із чутливістю даних, важливістю операції та історією взаємодій, внаслідок чого формують рішення щодо посилення або послаблення механізмів конфіденційності й цілісності. Для fog-вузлів,

що працюють у реальному часі, додатково актуальною є вимога збереження доступності та передбачуваності затримок навіть під дією атак, що зумовлює потребу спільного розгляду Threat Level, ресурсних обмежень та параметрів якості сервісу. Відповідність класів загроз типам адаптацій криптопрофілю fog-вузла наведено у табл. 2.4.

Таблиця 2.4 – Відповідність класів загроз типам адаптацій криптопрофілю fog-вузла

Клас загрози (Цілі безпеки)	Тип адаптації криптопрофілю	Стан енергоресурсів	Опис механізму адаптації та вплив
Конфіденційність (Confidentiality)	Посилення шифрування	Високий / Номінальний	Перехід від полегшених алгоритмів (напр., Simon/Speck) до стійких (AES-256) або збільшення кількості раундів перетворень.
Цілісність (Integrity)	Посилення автентифікації	Високий / Номінальний	Використання складніших хеш-функцій (SHA-3 замість SHA-1) або цифрових підписів з довшими ключами (ECC-384).
Доступність (Availability)	Зміна політик пріоритизації	Середній / Обмежений	Впровадження механізмів відсікання надлишкових запитів (Rate Limiting) та пріоритетна обробка критичних трафікових потоків для запобігання DoS-атакам.
Живучість (Survivability)	Спрощення профілю безпеки	Низький (Критичний)	Перехід на мінімально допустимі параметри безпеки (напр., зменшення частоти оновлення ключів, використання HMAC з меншим вікном) для максимізації часу роботи вузла.
Конфіденційність / Живучість	Динамічне керування довжиною ключа	Варіативний	Адаптація довжини ключа відповідно до залишкової ємності батареї: Erem→KeyLength.
Доступність / Цілісність	Адаптивна частота перевірки цілісності	Обмежений	Збільшення інтервалів між перевірками цілісності прошивки або даних під час падіння рівня заряду нижче порогу Tenergy.

Загрози, орієнтовані на конфіденційність (перехоплення, несанкціонований доступ, витік даних), за високих значень Threat Level вимагають переходу до криптографічних профілів підвищеної стійкості (збільшення довжини ключа,

посилення автентифікації, увімкнення додаткового шифрування на рівні застосунків), навіть якщо це збільшує затримку та енергоспоживання. Для загроз цілісності (ін'єкція фальшивих даних, модифікація пакетів, replay-атаки) доцільним є посилення механізмів автентифікації, контролю цілісності та журналювання, тоді як рівень шифрування можна залишити незмінним, якщо конфіденційність не є домінувальною вимогою. Загрози доступності (DoS/DDoS, flooding) вимагають адаптації політик пріоритизації трафіку, відкидання низькопріоритетних запитів, обмеження частини сервісів і, за потреби, спрощення криптопрофілів для критичних сервісів із метою збереження живучості та дотримання обмежень щодо затримок.

В енергообмежених fog-сценаріях Threat Level має бути безпосередньо пов'язаний із адаптацією криптографічного профілю з урахуванням залишкового заряду батареї, оскільки надмірне посилення захисту за високого рівня загроз здатне знизити живучість вузла через швидке виснаження акумулятора або перевантаження CPU. У QoS-орієнтованих моделях керування ресурсами fog-вузлів рішення про розміщення задач, використання offloading-у до хмари та активацію окремих механізмів безпеки пов'язують із поточним станом вузлів і вимогами до якості обслуговування. У межах цієї роботи такий стан інтерпретовано через показник Threat Level, який дає змогу узгоджувати вимоги до конфіденційності, цілісності й доступності з обмеженнями енергетичної ефективності.

2.2.6 Рівень загроз у задачі керування

Threat Level T_k у запропонованій постановці є компактним керувальним сигналом, який трансформує багатовимірну інформацію про стан безпеки fog-вузла до скалярної величини, придатної для використання в адаптивному механізмі вибору криптопрофілю. Ризиковий показник застосовують для прийняття рішень щодо повного дозволу, часткового дозволу або відмови в доступі, а також для вибору рівня захисту, що є прямим аналогом використання Threat Level як вхідного параметра цільової функції. Для fog-вузла, який є контрольованим об'єктом в задачі енергоорієнтованого захисту, політика керування (цільова функція) має враховувати

не лише поточний Threat Level, але й заряд батареї та навантаження, щоб уникати надто агресивних реакцій, які зумовлюють небажані витрати енергії. Функцію керування криптопрофілем можна подати у вигляді:

$$u_k = \pi(\bar{x}_k, T_k), \tag{2.9}$$

де u_k – обраний криптопрофіль або режим захисту, $\bar{x}_k$ – вектор станів, що містить заряд батареї, завантаження CPU та критичність поточних сервісів. Застосування надто жорстких реакцій (наприклад, повний перехід у максимально захищений режим під час кожного підвищення оцінки загроз) створює додаткові ризики через збільшення затримок, зниження пропускної здатності та підвищене енергоспоживання. Тому цільова функція має враховувати обмеження на частоту перемикань профілів, гістерезисні зони та пріоритизацію критичних сервісів із позиції живучості.

Вимоги до динамічних властивостей Threat Level охоплюють відсутність надмірної чутливості до одиничних подій (фільтрація шуму), плавне зростання в разі накопичення ознак атаки та можливість швидкого зниження після припинення атакувального впливу. Доцільно використовувати нечіткі та ймовірнісні методи для стабілізації Threat Level, що дає змогу уникати явища chattering – частих перемикань режимів захисту за умов малих коливань вхідних ознак. Для енергообмежених fog-вузлів поодинокі стрибки Threat Level, які спричиняють часті зміни криптопрофілю, миттєво відображаються на енергоспоживанні і можуть зумовити нераціональне використання енергоресурсів, тому в моделі доцільно застосовувати згладжувальні фільтри, гістерезисні пороги та адаптивне налаштування параметра α у формулі (2.8).

Стабільність керування ресурсами в умовах змінного навантаження забезпечують саме внаслідок згладжених оцінок стану системи, які не реагують на окремі піки, але чутливі до змін тренду. Аналогічні вимоги накладають на Threat Level як на керувальну змінну в механізмі адаптації криптопрофілю: надмірна інерційність спричинить запізніле реагування на атаки, тоді як надмірна чутливість – нестабільність режимів захисту й підвищене енергоспоживання. Компроміс між цими

крайнощами має бути відображений у виборі параметрів $\Phi(\cdot)$, α та структурі політики $\pi(\cdot)$, що буде розвинено в подальших підрозділах.

2.3 Модель якості криптографічного захисту

Поняття «якості» криптографічного профілю для fog-вузла в IoT-системі пов'язано з комплексом ресурсних витрат, зумовлених застосуванням конкретної конфігурації алгоритмів шифрування, цілісності та автентифікації, а також вибором параметрів ключів і режимів роботи. У працях з енергоусвідомленого захисту для IoT та туманних обчислень (fog computing) доведено, що навіть за умови фіксованої інтенсивності трафіку вибір криптографічного набору (cipher suite) здатен суттєво змінювати час оброблення, пропускну здатність та енергоспоживання вузла, що в умовах обмежених енергетичних ресурсів безпосередньо впливає на життєвий цикл системи [70]. Для fog-вузлів на базі архітектури ARM, які обслуговують множину ресурсно-обмежених сенсорних пристроїв, така «якість» має багатовимірний характер і охоплює щонайменше три основні компоненти: накладні витрати на процесорні ресурси (CPU-time), внесок у затримку оброблення й передавання (latency) та додаткові енергетичні витрати, що відображаються на стані заряду батареї (State of Charge).

Формально криптографічний профіль позначають як $P_i \in P$, де множина P описує фіксований набір допустимих конфігурацій (наприклад, профілі типу Safe, Balanced, Max, Survival, які відповідають різним рівням криптографічної «сили» та ресурсомісткості). Для кожного профілю P_i визначають тривимірний вектор вартості $\bar{V}(P_i)$:

$$\bar{V}(P_i) = \left[C_{cpu}(P_i), C_{lat}(P_i), C_{energy}(P_i)\right]^T, \tag{2.10}$$

де $C_{cpu}(P_i)$ характеризує відносні накладні витрати процесора, $C_{lat}(P_i)$ – додатковий внесок у затримку оброблення/передавання, а $C_{energy}(P_i)$ – відносну

енергетичну вартість профілю P_i порівняно з базовою конфігурацією. Енергетичну поведінку fog-вузлів доцільно моделювати через залежність споживаної потужності від завантаження CPU та інтенсивності мережевого обміну. На основі цих багатовимірних оцінок формують узагальнені показники якості сервісу, які відображають компроміс між затримками, пропускною здатністю та енергоспоживанням. Зазначене узагальнення забезпечує можливість оперувати криптографічним профілем як об’єктом керування, а не лише як статично заданою конфігурацією протоколу.

Взаємозв’язок між криптопрофілем та накладними витратами на процесорні ресурси визначають структурою криптографічних операцій та їхньою частотою. Кожен профіль P_i задає конкретний набір параметрів: тип алгоритму (симетричний шифр, асиметричний алгоритм, алгоритм узгодження ключів), довжину ключів, кількість раундів, режим шифрування (наприклад, лічильник з автентифікацією Галуа (Galois/Counter Mode, (GCM)), режим лічильника (Counter Mode, (CTR)), ChaCha20-Poly1305), а також наявність або відсутність додаткових механізмів автентифікації та цілісності. За умови фіксованої інтенсивності трафіку λ (кількість пакетів за одиницю часу) профіль із ресурсомісткішими алгоритмами породжує більшу кількість елементарних операцій (множення над скінченними полями, модульні піднесення до степеня, операції на еліптичних кривих), що збільшує середнє завантаження CPU fog-вузла. Відносні накладні витрати процесора для профілю P_i описують функцією $C_{cpu}(P_i)$:

$$C_{cpu}(P_i) = \frac{1}{N}\sum_{j=1}^{N} t_{cpu}(P_i, j)\,, \tag{2.11}$$

де $t_{cpu}(P_i, j)$ – час оброблення j-го блока або пакета даних у профілі P_i, а N – кількість блоків, що розглядають у нормувальному вікні.

Для сценаріїв із профілями типу Safe, Balanced, Max, Survival доцільно інтерпретувати $C_{cpu}(P_i)$ в нормованих одиницях, наприклад, як відносне

завантаження процесора в діапазоні [0,1] відносно базового режиму без криптографічного захисту. У такому разі ресурсомісткі профілі (Max) набувають значень $C_{cpu}(P_{Max})$, що суттєво перевищують параметри профілів Safe або Survival, які спираються на полегшені алгоритми та коротші ключі. Для fog-вузла на базі ARM SoC це дає змогу пов'язати криптографічну конфігурацію з виміряними метриками завантаження процесорних ядер, наприклад середнім відсотком використання обчислювальних ресурсів протягом фіксованого інтервалу часу.

Перехід від навантаження на CPU до затримок оброблення та наскрізних (end-to-end) затримок сервісів описують із застосуванням моделей масового обслуговування типу M/M/1 або G/G/1, які застосовують для оцінювання часу обслуговування та чергування запитів у fog- та edge-інфраструктурах. Для fog-вузла, який обробляє вхідний потік пакетів з інтенсивністю λ та має ефективну інтенсивність обслуговування $\mu(P_i)$, що залежить від криптопрофілю P_i, середній час перебування пакета в системі $T_{sys}(P_i)$ для простої моделі M/M/1 задають як:

$$T_{sys}(P_i) = \frac{1}{\mu(P_i) - \lambda}, \tag{2.12}$$

за умови стабільності системи $\lambda < \mu(P_i)$. Інтенсивність обслуговування $\mu(P_i)$ є обернено пропорційною до $C_{cpu}(P_i)$, оскільки збільшення середнього часу шифрування пакета безпосередньо зменшує кількість пакетів, які вузол здатен обробити за одиницю часу. Отже, зростання $C_{cpu}(P_i)$ спричиняє зростання середньої затримки $T_{sys}(P_i)$, що є особливо критичним для застосунків, чутливих до затримок.Відносний внесок криптопрофілю в затримку описують коефіцієнтом $C_{lat}(P_i)$:

$$C_{lat}(P_i) = \frac{T_{sys}(P_i)}{T_{sys}(P_0)}, \tag{2.13}$$

де $T_{sys}(P_0)$ – середня затримка в базовій конфігурації без шифрування або з мінімальним захистом.

Такий підхід дає змогу коректно порівнювати профілі між собою у відносній шкалі, тобто незалежно від конкретних абсолютних значень затримки, які суттєво залежать від архітектури fog-вузла, мережевого стека та характеру трафіку. У роботах щодо безпечного передавання даних у мережах із низьким енергоспоживанням і втратами (Low-Power and Lossy Networks, (LLN)) у контексті інтеграції з fog-інфраструктурою, доведено, що використання традиційних протоколів захисту з ресурсомісткими криптографічними та сигналізаційними процедурами (наприклад, повноцінних TLS/DTLS-рукопотискань) зумовлює суттєве збільшення обчислювальних накладних і трафікових витрат, що помітно збільшує end-to-end затримку, особливо в умовах високого навантаження й обмежених ресурсів LLN-вузлів. Це мотивує розроблення полегшених безпекових рішень, які зменшують кількість криптографічних операцій та службових повідомлень для підтримання прийнятної затримки у fog-орієнтованих сценаріях [71].

Зв’язок між навантаженням на CPU та енергоспоживанням fog-вузла розглядають в енергетичних моделях, де потужність споживання $P(t)$ подають як суму базової компоненти та компоненти, пропорційної завантаженню обчислювальних ресурсів. Для fog-вузла на базі ARM SoC спрощену модель подають у вигляді:

$$P(P_i, t) = P_{idle} + k_{cpu} \cdot U_{cpu}(P_i, t), \tag{2.14}$$

де P_{idle} – потужність у режимі простою, k_{cpu} – коефіцієнт, що відображає приріст потужності з ростом завантаження CPU, а $U_{cpu}(P_i, t)$ – миттєве завантаження процесора за використання профілю P_i.

У практичних експериментах для fog-шлюзів підтверджено, що перехід від менш ресурсомістких криптографічних протоколів до більш інтенсивних спричиняє суттєве зростання інтегральної енергії, спожитої на одиницю оброблених даних [72].

Вплив вибору криптопрофілю на стан заряду батареї описують рівняннями балансу заряду, де швидкість зміни заряду є функцією споживаного струму. Для дискретної часової моделі з кроком Δt еволюцію заряду батареї подають як:

$$SB_{k+1} = B_k - \frac{I(P_i, k) \cdot \Delta t}{C_{batt}}, \tag{2.15}$$

де $I(P_i, k)$ – струм, споживаний fog-вузлом у профілі P_i, а C_{batt} – номінальна ємність батареї. Струм $I(P_i, k)$ є функцією що зростає від потужності $P(P_i, k)$, отже, зростання потужності внаслідок вибору ресурсомісткого профілю спричиняє прискорене зменшення заряду батареї в часі. Для оцінювання відносної енергетичної вартості криптопрофілю вводять коефіцієнт $C_{energy}(P_i)$:

$$C_{energy}(P_i) = \frac{E(P_i)}{E(P_0)}, \tag{2.16}$$

де $E(P_i)$ – енергія, витрачена fog-вузлом на оброблення заданого обсягу даних у профілі P_i, а $E(P_0)$ – енергія для базової конфігурації. Такий підхід використовують для порівняння різних протоколів і параметрів мережевого стека з огляду на їхній внесок у життєвий цикл системи.

Комплексну модель зв’язку «профіль → CPU → latency → енергія/заряд батареї» базують на поєднанні залежностей (2.1)–(2.16) у єдиній структурі, де для кожного профілю P_i зафіксовано три взаємопов’язані коефіцієнти $C_{cpu}(P_i)$, $C_{lat}(P_i)$ та $C_{energy}(P_i)$. Такі коефіцієнти визначають експериментально шляхом вимірювання часу оброблення блоків даних, середнього завантаження CPU, затримки пакетів та енерговитрат для фіксованих сценаріїв трафіку і конфігурацій fog-вузла. Застосування цієї моделі у fog-інфраструктурах IoT дає змогу порівнювати криптографічні профілі не лише в термінах криптографічної стійкості, а й у термінах ресурсомісткості, що є критичним для енергообмежених сценаріїв мобільного або автономного електроживлення. Для забезпечення подальшої інтеграції моделі в цільову функцію

багатокритеріального вибору доцільно нормувати коефіцієнти C_{cpu}, C_{lat} та C_{energy} до інтервалу [0,1], використовуючи мінімальні та максимальні значення, отримані з вимірювань або з літератури. Для кожного виміру вартості вводять операції нормалізації, наприклад:

$$\widehat{C_{energy}}(P_i) = \frac{C_{energy}(P_i) - \min(C_{energy})}{\max(C_{energy}) - \min(C_{energy})}, \tag{2.17}$$

аналогічно для $\widehat{C_{cpu}}(P_i)$ та $\widehat{C_{lat}}(P_i)$. Отримані нормовані значення безпосередньо додають в інтегральний критерій якості J, який враховуватиме різні аспекти якості криптографічного профілю в єдиній шкалі.

Безпекову складову «якості» криптопрофілю формалізують через умовний показник рівня безпеки (Security Score) $S(P_i)$, який характеризує рівень криптографічної стійкості профілю за сукупністю параметрів. Для fog-інфраструктур додатково враховують наявність взаємної автентифікації між вузлом та хмарою, захист метаданих та забезпечення конфіденційності трафіку між сенсорами, fog-вузлами та хмарними сервісами. Узагальнено security score можна подати як:

$$S(P_i) = f_s(K_{len}, A_{type}, M_{mode}, Auth_{mech}), \tag{2.18}$$

де K_{len} – вектор довжин ключів для симетричних і асиметричних алгоритмів, A_{type} – типи використаних криптопримітивів (AES, ChaCha20, ECC тощо), M_{mode} – режими роботи шифрів і протоколів, а $Auth_{mech}$ – набір механізмів автентифікації та цілісності. Шкали рівнів безпеки або сертифікаційні рівні для IoT-об'єктів інтерпретують як дискретні значення для функції $S(P_i)$, наприклад, у вигляді сертифікатів рівня безпеки (Security Level Certificates) для різних класів пристроїв. Підвищення security score зазвичай супроводжують зростанням $\widehat{C_{cpu}}(P_i)$, $\widehat{C_{lat}}(P_i)$ і $\widehat{C_{energy}}(P_i)$, що відображає класичний компроміс між рівнем захисту та ресурсомісткістю.

Важливо, що Security Score є частиною векторної цільової функції, у якій виграш у безпеці протиставляють зростанню ресурсних витрат. Для задачі керування криптографічними профілями в енергообмежених fog-вузлах це означає потребу узгодження шкал для $S(P_i)$ та нормованих коефіцієнтів $\widehat{C_{cpu}}$, $\widehat{C_{lat}}$, $\widehat{C_{energy}}$, щоб забезпечити можливість формування інтегрального критерію якості J та формулювання багатокритеріальної задачі вибору. Для енергоефективного налаштування протоколів і параметрів безпеки в fog-середовищах доцільно використовувати узагальнені показники якості, що поєднують безпекові характеристики з енергетичними та часовими. У такій постановці рівень захисту аналізують разом із показниками енергоспоживання та затримки як компоненти одновимірного інтегрального критерію якості J або багатовимірного вектора якості сервісу.

Статичний вибір одного криптографічного профілю для fog-вузла є неефективним в умовах, коли заряд батареї, рівень загроз (Threat Level) і навантаження на вузол істотно змінюються в часі. Статичні політики захисту не враховують динаміки ресурсів, стану мережі та контексту загроз, що спричиняє або надмірне споживання енергії та обчислювальних ресурсів в умовах низького рівня ризику, або недостатній рівень захисту в умовах зростання загрози й обмеженого ресурсу. Для fog-вузла на батарейному живленні, що обслуговує критичні сервіси, фіксований вибір максимально ресурсомісткого профілю безпеки за умов низького Threat Level є прикладом ситуації, коли політика зумовлює невиправдано високі енергетичні витрати і скорочення часу автономної роботи без пропорційного практичного виграшу в захищеності порівняно з адаптивним підходом [73].

Іншим характерним сценарієм є ситуація, коли Threat Level є високим, але заряд батареї вже наблизився до критичних значень. Використання профілю Max у такому стані забезпечує підвищений security score, однак супроводжується істотним збільшенням $C_{energy}(P_i)$ та $C_{cpu}(P_i)$, що пришвидшує подальше падіння заряду батареї і здатне зумовити передчасне вимкнення fog-вузла, а отже – втрату сервісу та недоступність навіть базових функцій захисту. Політики самоадаптивної безпеки в IoT/Fog-сценаріях мають спиратися як на поточну оцінку загроз, так і на прогнозовану

зміну ресурсів, у тому числі заряду батареї. Це дає змогу уникнути жорстких статичних налаштувань, які або перевитрачають енергію, або залишають систему недостатньо захищеною в умовах довготривалої експлуатації.

Нездатність статичних політик збалансувати компроміс між безпекою та ресурсами виявляється також у тому, що такі політики фіксують співвідношення між $S(P_i)$ і $\bar{V}(P_i)$ незалежно від змінних параметрів заряду батареї, Threat Level і навантаження. Внаслідок цього виникають два характерні режими: over-protection, коли високий security score досягають ціною неприйнятних енергетичних та часових витрат, та under-protection, коли економію ресурсу досягають шляхом зниження security score до рівнів, що не відповідають актуальному профілю загроз. Для багатьох класів застосувань, зокрема в розподілених архітектурах fog computing, відсутні усталені моделі, які б системно поєднували безпекові та енергетичні аспекти в єдиному інтегральному критерії якості J.

Потребу моделі «якості» криптографічного захисту в задачі керування криптопрофілями fog-вузла зумовлено тим, що заряд батареї, Threat Level та навантаження належать до динамічних змінних, які необхідно враховувати під час прийняття рішень. Енергомодель у поєднанні з показниками безпеки і затримки формує основу для побудови цільової функції, яка відображає компроміс між виграшем у безпеці та витратами ресурсів. Багатокритеріальний вибір у межах туманних обчислень ґрунтується на векторі показників (енергія, затримка, надійність), який перетворюють на єдину скалярну цільову функцію за допомогою ваг або системи обмежень. З позицій керування криптографічними профілями fog-вузла модель «якості» має відображати можливість ухвалення рішень про зміну профілю на основі спостережуваних значень заряду батареї, Threat Level, поточного навантаження та прогнозованих ресурсних витрат. Така модель дає змогу оцінити механізми режимів керованої деградації, у яких зниження рівня безпеки відбувається відповідно до зменшення ресурсу батареї та зростання навантаження, зберігаючи мінімально допустимий Security Score для актуального профілю загроз.

2.4 Інтегральний критерій якості та формулювання задачі вибору

Інтегральна оцінка якості керування криптографічним захистом fog-вузла є ключовим елементом математичної постановки задачі енергобезпекового компромісу, оскільки енергоспоживання, затримка та рівень безпеки формують суперечливі вимоги до конфігурації криптографічних профілів в IoT- та fog-середовищі. Узагальнення цих вимог у скалярний інтегральний критерій якості J забезпечує можливість формального порівняння політик керування та формулювання задачі збалансованого керування адаптивним механізмом захисту на базі fog-вузлів з архітектурою ARM, включно з платформами класу Raspberry Pi.

2.4.1 Необхідність інтегрального критерію якості

Аналіз енергоспоживання в ізоляції від затримок і рівня безпеки зумовлює переоцінку стратегій, які мінімізують тільки спожиту енергію, але істотно погіршують якість сервісу або знижують криптостійкість каналів зв’язку. У роботах з оцінювання продуктивності криптографічних протоколів на fog- та mist-шлюзах доведено, що вибір криптографічних механізмів і пов’язаних із ними процедур (тип асиметричної криптографії, схема автентифікації, кількість раундів встановлення з'єднання) має істотний вплив на час встановлення захищеної сесії, пропускну здатність і енергоспоживання вузла, що є критичним для частини IoT-сценаріїв реального часу. Зокрема, порівняльні експерименти на fog- та mist-пристроях показують, що належним чином підібрані ECC-набори шифрів для TLS дають змогу одночасно зменшити енергоспоживання й підвищити ефективну пропускну здатність порівняно з традиційними RSA-профілями, що підкреслює важливість узгодженого вибору протоколу та параметрів безпеки з вимогами до затримки [74]. Розгляд лише криптографічної стійкості без урахування енергетичних ресурсів спричиняє вибір профілів із максимальною криптографічною стійкістю (довгі ключі, складні протоколи встановлення з'єднання, розширена автентифікація), які для батарейних

fog-вузлів на кшталт Raspberry Pi 4/5 породжують надмірне навантаження на процесор і прискорене зниження заряду акумулятора.

Одновимірні критерії (наприклад, «мінімум споживаної потужності» або «мінімум затримки») недостатньо адекватні для систем із жорсткими обмеженнями як на живучість, так і на якість сервісу, зокрема в контексті безпроводових сенсорних мереж та мобільних edge-вузлів. Для криптографічних підсистем IoT-пристроїв це виявляється в тому, що скорочення обчислювальної складності алгоритмів заради економії енергії здатне знизити рівень захисту нижче від регуляторних або галузевих вимог, зокрема щодо стійкості до сучасних атак на протоколи обміну ключами.

У контексті fog-обчислень інтегральний підхід є особливо необхідним через те, що fog-вузли одночасно виконують роль проміжних обчислювальних елементів і вузлів безпеки (термінація TLS/DTLS, реалізація OSCORE, проксі-шифрування), що накладає обмеження як на пропускну здатність, так і на енергетичний профіль пристрою. Окремий розгляд тільки затримок зумовлює пріоритизацію профілів із мінімальною криптографічною складністю, тоді як аналіз тільки рівня захисту стимулює вибір більш криптографічно складних схем, що суперечить довготривалій автономній роботі fog-вузла. Поняття «енергобезпековий компроміс» для IoT- та fog-систем належить до класу багатокритеріальних задач, для розв'язання яких застосовують узагальнені критерії якості, що агрегують окремі показники в єдину функцію, придатну для процедур оцінки адаптивного вибору профілів.

У сучасних підходах до енергоощадного керування обчислювальними навантаженнями на edge-обладнанні застосовано формалізацію у вигляді багатокритеріального збалансування із застосуванням цільових функцій, що поєднують затримку, енергоспоживання та точність/якість обслуговування. Для задачі керування криптографічними профілями fog-вузла аналогічна стратегія полягає у введенні інтегрального критерію якості J, що відображає сумарну «якість» обраної політики у розрізі енергетичних витрат, затримок і невідповідності до цільового рівня безпеки. Така функція дає змогу порівнювати різні політики керування, навіть якщо одна з них покращує затримку ціною погіршення енергоефективності, а інша –

навпаки, і забезпечує можливість будувати алгоритми адаптивного вибору криптопрофілів на основі мінімізації J.

2.4.2 Формальне введення інтегрального критерію якості

Інтегральний критерій якості J вводять як скалярну міру якості політики керування криптографічними профілями fog-вузла, що агрегує три базові нормовані показники: затримку сервісу, енергетичні втрати та недостатній рівень безпеки. Для реалізації підходу в задачі керування криптографічним захистом визначено трикомпонентний вектор нормованих показників $\bar{V} = [L, P, S]^T$, де кожен компонент належить до інтервалу $[0,1]$ і характеризує негативну сторону відповідного аспекту (затримки, енергоспоживання, недостатність безпеки).

Показник L (Latency Penalty) інтерпретують як нормовану міру деградації якості сервісу, що відображає затримку оброблення запитів користувачів або IoT-пристроїв порівняно з гранично допустимим значенням для відповідного класу застосувань. Для fog-вузлів, що обслуговують сценарії, чутливі до затримок (керування виконавчими механізмами, реальний час), зростання L свідчить про наближення системи до недопустимого рівня затримок, зумовленого вибором надто «важких» криптографічних профілів або перевантаженням процесора. Показник P (Power/Energy Penalty) описує нормовану інтенсивність споживання енергії криптографічною підсистемою fog-вузла або нормовану швидкість зниження заряду акумулятора, що є критичним для мобільних та автономних сценаріїв. Показник S (Security Penalty) репрезентує нормовану «недостатність» безпеки, наприклад відносне відхилення фактичного рівня безпеки (security score) від бажаного рівня, який задано політиками безпеки або галузевими стандартами. Для забезпечення порівнянності внеску різних складових доцільно нормувати кожен показник до інтервалу $[0,1]$, де 0 відповідає найкращому стану щодо відсутності втрат і ризиків (мінімальна затримка, мінімальна швидкість розряду, максимальний рівень безпеки), а 1 описує найгірший допустимий стан у заданому діапазоні експлуатації fog-вузла. За таких умов інтегральний критерій подано у вигляді зваженої суми:

$$J = w_L \cdot L + w_P \cdot P + w_S \cdot S, \tag{2.19}$$

де w_L, w_P, w_S – вагові коефіцієнти, що відображають пріоритетність затримки, енергетичної складової та безпеки відповідно, за таких умов $w_L + w_P + w_S = 1$. Аналогічні лінійні критерії застосовують у роботах з оптимізації параметрів криптографічних прискорювачів та полегшених (lightweight) алгоритмів для IoT-пристроїв, де ваги використовують для налаштування компромісу між продуктивністю, енергоспоживанням і стійкістю.

Формула (2.19) є основою для побудови підходів до адаптивного керування, у яких задача синтезу політики переходів між криптопрофілями сформована як задача мінімізації цільового критерію J за наявності обмежень на допустимі значення затримок та заряду батареї. Вагові коефіцієнти можуть бути статичними (заданими на етапі проєктування) або динамічно налаштовуваними відповідно до поточного режиму роботи fog-вузла (наприклад, зміна пріоритетів в умовах критично низького заряду батареї або підвищеного рівня загроз).

2.4.3 Нормалізація окремих показників затримки, енергії та безпеки

Коректна нормалізація показників L, P, S до інтервалу $[0,1]$ є необхідною умовою для того, щоб їхня зважена сума у (2.19) адекватно відображала внесок кожної складової до інтегральної «якості» політики керування. Оцінювання енергоспоживання та продуктивності алгоритмів на ARM-платформах, нормалізацію можна здійснювати відносно гранично допустимих або експериментально отриманих екстремальних значень метрик, наприклад максимального спостереженого енергоспоживання чи затримки. Для fog-вузла необхідно враховувати як теоретичні обмеження, визначені вимогами до сервісів, так і фактичні характеристики, отримані в процесі вимірювань на цільовому стенді.

Для енергетичної складової P практика edge- та IoT-досліджень застосовує нормування відносно максимально допустимої швидкості розряду акумулятора або

мінімально прийнятного часу автономної роботи, визначеного для конкретного сценарію. Наприклад, значення $P = 0$ відповідає режиму, у якому миттєва потужність, споживана криптографічними операціями fog-вузла, не перевищує базового порога, пов'язаного з комфортним енергетичним режимом, тоді як $P = 1$ відображає граничний режим, за якого швидкість розряду зумовлює неприйнятне скорочення часу автономної роботи.

Для затримок показник L доцільно нормувати відносно гранично допустимого часу відповіді сервісу, визначеного вимогами застосунку (наприклад, обмеженнями 3GPP URLLC або вимогами до керування фізичними процесами). $L = 0$ відповідає ситуації, коли середня затримка криптографічних операцій і пов'язаних із ними протокольних процедур є значно нижчою за порогове значення, тоді як $L = 1$ означає досягнення або перевищення максимально допустимої затримки.

Показник безпеки S нормують відносно цільового або максимально можливого рівня безпеки (security score), що базується на формальних моделях оцінювання криптостійкості (розмір простору ключів, стійкість до конкретних класів атак) чи композитних метриках, запропонованих у стандартах і оглядах з IoT-безпеки. Для fog-вузла, що реалізує кілька криптопрофілів (різні довжини ключів, алгоритми, протоколи), задають шкалу, де $S = 0$ відповідає профілю, який повністю задовольняє цільові вимоги (наприклад, мінімальний рекомендований рівень безпеки для певного класу застосувань), а $S = 1$ – найменш захищеному профілю в наборі, який ще допускається політикою системи.

Нормалізація на основі теоретично заданих меж має узгоджуватися з експериментальними даними, отриманими на fog-стенді, оскільки реальні характеристики енергоспоживання та затримок криптографічних операцій залежать від конкретної реалізації, конфігурації мікропроцесора та операційної системи. У дослідженнях, виконаних на Raspberry Pi та подібних одноплатних платформах, показано, що навіть для одного й того самого криптографічного алгоритму різні варіанти реалізації (зокрема, оптимізовані або модифіковані за числом раундів, а також реалізації з використанням специфічних бібліотек чи апаратних можливостей) демонструють помітно відмінні профілі часу виконання, завантаження процесора та

енергоспоживання. Це вимагає калібрування параметрів моделей та нормувальних функцій продуктивності й енерговитрат на основі емпіричних вимірювань для конкретних fog-/edge-платформ [75]. Правильна нормалізація забезпечує порівнянність внеску L, P та S у критерії J та запобігає домінуванню одного параметра через відмінність масштабів вимірювання. Приклади формул нормалізації наведено у табл. 2.5.

Таблиця 2.5 – Приклади формул нормалізації для L, P, S та способи визначення меж

Показник	Формула нормалізації	Опис визначення граничних значень
Затримка (L)	$L = \frac{T_{cur} - T_{min}}{T_{max} - T_{min}}$	T_{max} задають вимогами QoS застосунку (напр., 50 мс для IIoT), T_{min} вимірюють на стенді без шифрування.
Енергія (P)	$P = \frac{E_{cur} - E_{idle}}{E_{max} - E_{idle}}$	E_{max} відповідає профілю Max за 100% завантаження CPU, E_{idle} – споживання вузла у режимі очікування.
Безпека (S)	$S = \frac{Sec_{target} - Sec_{cur}}{Sec_{target} - Sec_{min}}$	Sec_{target} – еталонний security score (напр., AES-256 + SHA-3), Sec_{min} – мінімально допустимий стандартом рівень.

2.4.4 Вибір вагових коефіцієнтів для різних сценаріїв

Вагові коефіцієнти w_L, w_P, w_S у (2.19) задають відносну важливість затримки, енергоспоживання та безпеки для конкретного сценарію експлуатації fog-вузла, що узгоджено з принципами багатокритеріального вибору в енергетично обмежених обчислювальних системах. Для IoT- та fog-середовища доцільно виділяти типові класи сценаріїв, серед яких виокремлюють збалансований (Balanced), безпеково-орієнтований (Security-oriented) та енергоорієнтований (Energy-oriented) режими. У збалансованому режимі пріоритети системи рівномірно розподіляють між підтриманням прийнятного рівня затримок, забезпеченням життєздатності пристрою внаслідок ощадливого використання енергії та дотримання нормативно заданого рівня безпеки.

У безпеково-орієнтованому сценарії параметр w_S набуває підвищеного значення, що відображає пріоритетність дотримання цільового рівня безпеки навіть за умов збільшення енергоспоживання й затримок. Подібні режими є характерними,

наприклад, для критичної інфраструктури або фінансових застосувань, де fog-вузол виступає в ролі шлюзу для підтвердження транзакцій, а компроміс із безпекою є неприйнятним. Енергоорієнтований сценарій, навпаки, підвищує вагу w_P та допускає деяке послаблення вимог до безпеки (через зниження довжин ключів, вибір легших алгоритмів, зменшення частоти криптографічних операцій), що є актуальним для мобільних fog-вузлів із електроживленням від акумулятора, де критичною є тривалість автономної роботи. Приклади налаштувань вагових коефіцієнтів наведено у табл. 2.6.

Таблиця 2.6 – Приклади налаштувань вагових коефіцієнтів для різних сценаріїв

Сценарій	Ваги (w_L, w_P, w_S)	Очікувана поведінка системи
Balanced (Збалансований)	$w_L = 0.33, w_P = 0.33, w_S = 0.34$	Рівномірний компроміс. У разі падіння заряду батареї або зростання затримки система плавно знижує криптопрофіль до середнього рівня.
Security-oriented (Безпековий)	$w_L = 0.2, w_P = 0.1, w_S = 0.7$	Пріоритет безпеки. Профіль залишається високим (Max/Safe) навіть в умовах дефіциту енергії, доки заряд не досягне критичного мінімуму.
Energy-oriented (Енергоощадний)	$w_L = 0.2, w_P = 0.6, w_S = 0.2$	Жорстка економія. Система швидко переходить на профіль Survival у разі найменших ознак розряду батареї для максимізації часу роботи.

Регулювання співвідношення між w_L, w_P, w_S дає змогу адаптувати поведінку системи до конкретного класу застосувань і поточного контексту роботи fog-вузла. Наприклад, у разі зменшення заряду батареї нижче певного порога адміністратор здатний збільшити w_P та зменшити w_S у межах допустимого рівня безпеки, змінюючи оцінку механізму керування. В адаптивних енергоощадних системах ваги розглядають як параметри, які налаштовують як на етапі проєктування, так і під час експлуатації, зокрема через інтерфейси керування політиками або автоматично на основі аналізу контексту та профілю навантаження.

Вибір вагових коефіцієнтів є задачею, яку вирішують на декількох рівнях: стратегічному (визначення класів застосувань і базових профілів), тактичному

(налаштування профілів для окремих сервісів або груп пристроїв) та операційному (динамічне коригування ваг у відповідь на зміну стану мережі, заряду батареї або рівня загроз). У цьому контексті інтегральний критерій якості J виступає узгоджувальним механізмом, що забезпечує єдину шкалу для оцінювання наслідків зміни параметрів керування та дає змогу адміністратору або надбудованій системі керування інтерпретувати вплив зміни ваг у термінах погіршення чи покращення енергобезпекового компромісу.

2.4.5 Формулювання задачі адаптивного керування

Математичну постановку задачі вибору політики керування криптографічними профілями fog-вузла базують на мінімізації цільового критерію J за наявності обмежень, що відображають вимоги до живучості та якості сервісів. У термінах теорії оптимального керування це відповідає побудові керувальної стратегії, яка мінімізує певний функціонал якості за наявності обмежень на стан системи. У розглядуваному випадку роль керування відіграє політика переходів між криптографічними профілями залежно від стану заряду батареї, поточного навантаження CPU, рівня загроз і заданих ваг у (2.19), а стан системи охоплює енергетичну та безпекову змінні.

Загальне формулювання задачі вибору має такий вигляд:

$$\min_{u_k \in U} J(\bar{u}_k) = w_L \cdot L(\bar{u}_k) + w_P \cdot P(\bar{u}_k) + w_S \cdot S(\bar{u}_k) \tag{2.20}$$

за умов:

$$S(u_k) \le S_{max_penalty}, B_k \ge B_{min}, L_{delay}(u_k) \le L_{max_delay}, \tag{2.21}$$

де $S_{max_penalty}$ – максимально допустима «нестача» безпеки (тобто мінімально допустимий рівень безпеки для обраного класу застосувань), B_{min} – порогове значення стану заряду батареї, нижче за яке fog-вузол має перейти до режиму

керованої деградації або вимушеної зупинки сервісів, L_{max_delay} – максимально допустима затримка для критичних сервісів. Подібна форма задачі є стандартною для аналізу компромісу між латентністю та енергією в edge-обчисленнях, де мінімізують зважену суму нормованих показників за наявності обмежень на QoS і ресурсні параметри.

Обмеження гарантують, що система не знижуватиме криптографічний профіль нижче від певного безпекового порога навіть у ситуаціях критично низького заряду батареї, зокрема якщо fog-вузол обробляє чутливі транзакції або керує критичними об'єктами. Аналогічно, затримка не може перевищувати L_{max_delay}, оскільки це спричинить порушення вимог до якості сервісу, визначених для латентно-чутливих застосувань. Для заряду батареї обмеження гарантує, що політика керування не приведе систему до ситуації, за якої fog-вузол не зможе забезпечувати ключові функції або коректно завершити захищені сесії.

Формалізація створює основу для застосування методів оптимізації та синтезу керування, як у класичному, так і в наближеному вигляді. У практичних реалізаціях fog-вузлів, де необхідна робота в реальному часі, знаходження глобального рішення за всім простором політик є недоцільним, проте критерій J і система обмежень успішно використовуються як локальна функція корисності для прийняття рішень щодо зміни криптопрофілю під час зміни стану системи.

2.4.6 Тип задачі та властивості розв'язку

Відповідно до теорії керування, сформульована задача належить до класу задач синтезу керувальної стратегії для динамічної системи з обмеженнями на стан і керування. Таку систему описують динамікою зміни заряду батареї, завантаження CPU, рівня загроз, а також логікою переходів між криптографічними профілями, що впливають на компоненти L, P, S і, відповідно, на J. З огляду на наявність невизначеностей (несталі профілі трафіку, випадкові зміни рівня загроз, варіативність енергоспоживання через температурні фактори), побудова строгого аналітичного розв'язку задачі знаходження глобально-ефективного рішення є складною.

У керуванні ресурсами в edge-обчисленнях широко застосовують наближені методи, включно з нечітким керуванням, евристичними алгоритмами та методами підкріплювального навчання, які розглядають формально задану функцію корисності як цільову функцію, але не прагнуть до аналітичного отримання глобального оптимуму. Для задачі енергоорієнтованого керування криптографічним захистом fog-вузла застосування нечіткого логічного контролера (Fuzzy Logic Controller, (FLC)) є доцільним, оскільки дає змогу формалізувати експертні правила типу «якщо заряд низький і рівень загроз невисокий, зменшити криптопрофіль» у вигляді бази правил із нечіткими множинами на множині лінгвістичних змінних (заряд батареї, Threat Level, завантаження CPU). У такій постановці інтегральний критерій якості J використовують як міру якості, яку оцінюють для порівняння політик на етапі проєктування та калібрування FLC, а також як орієнтир для налаштування параметрів нечітких множин і вагових коефіцієнтів.

Властивості розв’язку такої задачі характеризують компромісом між строгістю оптимальності та обчислювальною складністю. Для fog-вузлів, що працюють у режимі реального часу, застосування полегшених, інтерпретованих моделей керування (на кшталт FLC) є більш практичним, ніж використання складних алгоритмів пошуку глобального максимуму функціонала, особливо в умовах енергетичних обмежень. Критерій J у цьому контексті виконує роль формалізованої бази, на основі якої задають вимоги до поведінки керувального механізму, але сам процес пошуку розв’язку реалізують наближеними методами, здатними працювати на ARM-базованих SoC з обмеженими ресурсами.

2.4.7 Інтерпретація значень інтегрального критерію якості для порівняння стратегій

Інтерпретацію значень інтегрального критерію якості J у контексті fog-вузла базують на тому, що нижчі значення J відповідають кращому компромісу між енергоспоживанням, затримками та безпекою за заданих вагових коефіцієнтів. У разі

керування криптопрофілями fog-вузла це означає, що політики, які забезпечують менші значення J, вважають кращими в сенсі енергобезпекового компромісу.

Прикладом є порівняння двох політик: політика А, яка агресивно знижує криптопрофіль під час зменшення заряду батареї для збереження енергії, та політика В, яка зберігає високий рівень безпеки до нижчих значень заряду батареї, але внаслідок зростання енергоспоживання і затримок. Для одного набору ваг, що відповідає збалансованому сценарію, політика А здатна мати нижче значення J в енергокритичних станах, тоді як в умовах нормального заряду батареї політика В може виявитися кращою через вищий рівень безпеки за помірних витрат енергії. Визначення J для кожної політики в типових робочих точках fog-вузла дає змогу систематизувати такі порівняння й уникнути суто якісних міркувань.

Цінність інтегрального критерію також полягає в можливості оцінювання ефекту запропонованих адаптивних методів порівняно зі статичними стратегіями, які використовують фіксований криптопрофіль незалежно від заряду батареї, рівня загроз або навантаження. В енергоефективному розгортанні алгоритмів на edge-платформах подібні критерії застосовують для демонстрації виграшу нових підходів у багатовимірному просторі показників (енергія/латентність/якість). У випадку криптографічного захисту fog-вузла інтегральний критерій якості J дає змогу формалізувати твердження про покращення «якісного» компромісу, наприклад шляхом зниження J внаслідок адаптивного вибору криптопрофілю за аналогічного або вищого рівня безпеки порівняно зі статичними стратегіями.

Висновки до розділу

У розділі здійснено математичну постановку задачі енергоорієнтованого керування криптографічним захистом fog-вузла в системах мобільного IoT. На основі проведеного формального опису fog-вузла як об’єкта керування ідентифіковано кілька змістовних обмежень наявних підходів. Зокрема, встановлено, що наявні адаптивні механізми захисту рідко використовують узгоджений вектор стану, який одночасно охоплює заряд батареї, навантаження, телеметрію якості обслуговування (QoS) та

індикатори загроз. Крім того, виявлено, що наявні моделі переважно не містять явно означеного поняття плавного зниження рівня захисту, коли вибір криптопрофілю в умовах деградації енергоресурсу реалізують не шляхом простого вимкнення механізмів захисту, а структурованою зміною профілю з контролем ризиків.

Розглянута класифікація загроз підтверджує, що fog-рівень одночасно є об'єктом мережевих, протокольних, ресурсоорієнтованих та даноцентричних атак, до того ж навантажувальні та DDoS-атаки відіграють особливу роль як тригери зростання рівня загроз і прискореного виснаження енергоресурсів. Доведено, що агрегований індикатор рівня загроз (Threat Level), який побудовано на основі телеметрії fog-вузла та метрик систем виявлення вторгнень (IDS) із застосуванням згладжувальних методів, дає змогу формалізувати поточний стан безпеки з урахуванням вимог щодо конфіденційності, цілісності, доступності (CIA) та живучості. Його застосовують як компактний керувальний сигнал для адаптації криптографічного профілю. Водночас аналіз літератури демонструє обмежену кількість підходів, у яких Threat Level явно інтегровано з енергетичними параметрами fog-вузла та використано у задачі спільного узгодження компромісу «енергія-затримка-безпека» (energy/latency/security) з урахуванням живучості й вимог щодо стабільності перемикань профілів.

З метою усунення цих недоліків запропонована модель «якості» криптографічного захисту, яка містить вектор ресурсних витрат та показник безпеки, створює базу для формування інтегрального критерію якості J, у якому енергетичні, часові та безпекові аспекти розглядають як взаємопов’язані компоненти єдиної задачі адаптивного керування. Для кожного криптопрофілю така модель забезпечує можливість еквівалентного порівняння різних конфігурацій захисту в координатах «ресурсомісткість – рівень безпеки», що є необхідною умовою для синтезу адаптивного механізму керування профілями fog-вузла.

Для оцінювання політик керування сформовано математичну модель інтегрального критерію якості J, яка агрегує нормовані показники затримки, енергоспоживання й рівня безпеки у зважену суму з відповідними ваговими коефіцієнтами. Доведено, що розгляд окремих показників відірвано один від одного

унеможливлює однозначне оцінювання ефективності стратегій, оскільки зміна криптопрофілю впливає одночасно на енергетичний профіль, латентність і криптостійкість, водночас ці ефекти є суперечливими. Запропонована схема нормалізації метрик і постановка задачі мінімізації цільового критерію за наявності обмежень на мінімально допустимий рівень безпеки, заряд батареї та максимально допустиму затримку формують формальну основу для подальшого синтезу механізмів керування.

Сформульоване завдання щодо багатокритеріального вибору та введення інтегрального критерію якості J задають теоретичну основу для розроблення безпосереднього методу керування. Оскільки аналітичний пошук глобального оптимуму в реальному часі є складним через ресурсні обмеження та високу динаміку середовища, отримані моделі обґрунтовують потребу застосування наближених алгоритмів. Зокрема, вони утворюють базис для побудови в подальшому розділі нечіткого логічного контролера (FLC), бази відповідних правил і набору криптопрофілів, які реалізують наближений пошук оптимуму, що забезпечує енергоорієнтоване адаптивне керування криптографічним захистом fog-вузла з підтримкою режимів плавного зниження рівня захисту.

3 МЕТОД ЕНЕРГООРІЄНТОВАНОГО АДАПТИВНОГО КЕРУВАННЯ

Функціонування туманних (fog) вузлів у розподілених системах IoT супроводжується жорсткими апаратними обмеженнями, насамперед щодо доступного енергетичного бюджету автономних джерел електроживлення. Використання статичних та ресурсомістких криптографічних протоколів спричиняє інтенсивне завантаження CPU обчислювача, що критично пришвидшує розряд батареї та знижує загальну живучість апаратної платформи. З огляду на це виникає науково-технічна задача синтезу замкненого контуру керування, здатного динамічно адаптувати обчислювальне навантаження криптографічного стека до поточного енергетичного стану пристрою.

Метою цього розділу є розроблення та теоретичне обґрунтування методу енергоорієнтованого адаптивного керування криптографічним захистом fog-вузлів. Для досягнення мети в розділі послідовно вирішують такі завдання: формування архітектурної концепції контуру моніторингу апаратних ресурсів, синтез нечіткого логічного контролера (FLC) типу Такагі-Сугено нульового порядку для вибору криптографічного профілю, а також алгоритмічна формалізація запропонованого методу для подальшої апаратної реалізації в ізольованому середовищі.

3.1 Концепція адаптивного фреймворку криптозахисту вузла туманних обчислень

Адаптивний менеджер безпеки (Adaptive Security Manager (ASM)) для fog-вузла реалізує енергоорієнтований контур керування криптографічним навантаженням, що динамічно узгоджує рівень захисту з енергетичним станом вузла та рівнем загроз (Threat Level) і реалізований на платформі апаратного моделювання (Hardware-in-the-Loop (HiL)) у Docker-оточенні.

3.1.1 Загальна архітектурна концепція фреймворку адаптивного керування на вузлі туманних обчислень

Ключовою властивістю ASM є контейнеризована реалізація на базі Docker, що забезпечує відтворюваність експериментів, переносимість між різними fog-платформами та можливість масштабування до кластерів fog-вузлів у вигляді мікросервісних fog-фреймворків. Підхід, за якого енергоорієнтовані механізми безпеки реалізують як окремий сервісний шар і відокремлюють від прикладних сервісів, узгоджується із сучасними тенденціями побудови енергообізнаних та мікросервісно організованих безпекових компонентів в IoT- та fog-інфраструктурах [76]. Блок-схему ASM-фреймворку наведено на рис. 3.1.

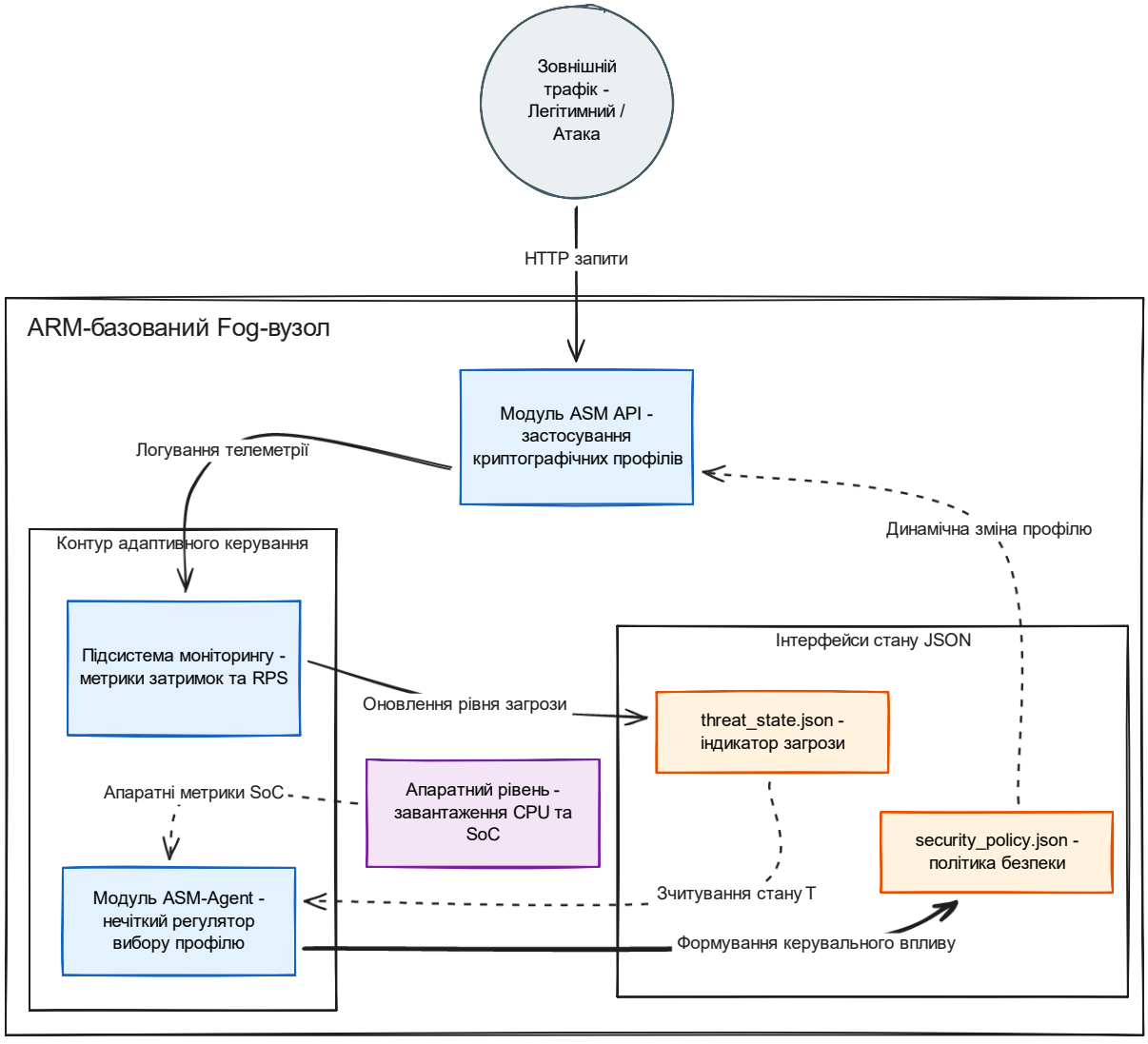

Рисунок 3.1 – Блок-схема ASM-фреймворку

У запропонованому ASM-фреймворку замкнений контур керування реалізують як послідовність етапів: моніторинг апаратних та сервісних метрик fog-вузла, оновлення інтегрального показника рівня загрози (Threat Level), обчислення нечіткого показника безпеки (Security Score) у FLC, вибір криптографічного профілю, застосування вибраного профілю в ASM API та логування результатів на кожному інтервалі керування. Така організація відповідає підходам до контекстно-залежного та ризик-орієнтованого керування доступом у Fog-IoT, де адаптивні агенти на fog-вузлах на основі моніторингу контексту обчислюють ризикові метрики й динамічно коригують політики доступу та безпеки [77] . Структурну схему контуру показано на рис. 3.2.

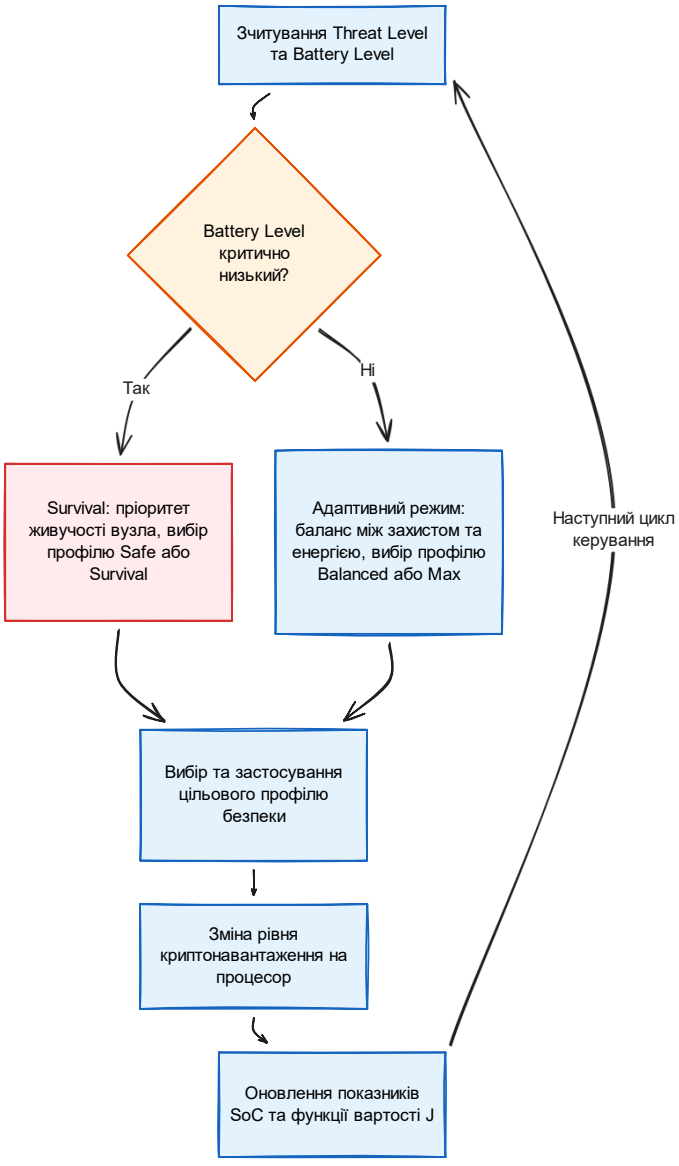

Рисунок 3.2 – Структурна схема контуру

3.1.2 Контур моніторингу стану вузла та введення рівня загроз

Моніторинг у компоненті ASM-Agent реалізовано на основі циклічного виконання з інтервалом, який задано в конфігурації (типово 10 с), у межах якого оцінюють віртуальний стан заряду батареї, завантаження CPU, температуру процесора, середню затримку оброблення запитів та актуальний Threat Level. Дані про поточне завантаження CPU та температуру отримують із застосуванням бібліотеки psutil, що створює можливість пов'язати обраний криптопрофіль із реальним апаратним навантаженням у конкретній конфігурації fog-вузла. З позиції апаратної електроніки, безперервний моніторинг температури є критично важливим для забезпечення живучості системи. У разі тривалого застосування ресурсомістких криптографічних примітивів (зокрема, у профілі Max) інтенсивне перемикання транзисторів спричиняє швидке зростання температури напівпровідникового переходу кристала (junction temperature). За досягнення критичних температурних порогів апаратний контролер живлення примусово активує механізм динамічного масштабування напруги та частоти – тепловий тротлінг (thermal throttling). Це фізичне явище супроводжується зниженням тактової частоти та напруги живлення ядра мікропроцесора для запобігання його тепловій деградації. Як наслідок, різко зростає час виконання процесорних інструкцій, що зумовлює неконтрольоване збільшення апаратної затримки оброблення пакетів (L_{norm}). Отже, температурний режим та апаратні обмеження кристала перетворюють адаптивне керування криптонавантаженням із суто програмної задачі на фізичну необхідність для стабільного функціонування пристрою.

Значення Threat Level у запропонованій архітектурі зберігають у файлі, який використовують як простий інтерфейс обміну між ASM-Agent та зовнішніми компонентами, зокрема системами виявлення вторгнень (IDS) або генераторами тестових сценаріїв HiL-стенду. Threat Level задають як безрозмірний показник, нормований до інтервалу $[0,1]$, де 0 відповідає номінальним умовам, а значення, близькі до 1, інтерпретують як високий рівень загрози (наприклад, інтенсивні DDoS-атаки чи виявлені аномалії трафіку). Така агрегація різнорідних сигналів безпеки в

єдину інтегральну оцінку узгоджується з рекомендаціями щодо побудови нормованих ризикових/загрозових показників для IoT-мереж і Fog-IoT систем, які використовують як вхід для контекстно-залежних політик доступу та адаптивного керування безпекою [78]. Перелік основних сигналів наведено в табл. 3.1.

Таблиця 3.1 – Перелік основних сигналів ASM-фреймворку

Назва сигналу / параметра	Джерело даних	Одиниці вимірювання	Періодичність оновлення
Рівень заряду (Battery Level)	Емпірична модель логічного розряду на базі апаратного завантаження CPU (бібліотека psutil), розраховується у asm_agent.py	Відсотки (%)	Дискретно, кожні 10 с (визначається параметром ASM_INTERVAL_SEC)
Рівень загрози (Threat Level)	Зовнішній генератор навантаження або система виявлення вторгнень (IDS), зчитується з файлу threat_state.json	Нормалізована безрозмірна величина (діапазон [0;1.0])	Дискретно, кожні 10 с (цикл опитування агента)
Оцінка безпеки (Security Score)	Вихідний сигнал нечіткого контролера (FLC) після етапу дефазифікації, модуль asm_agent.py	Безрозмірна числова величина (діапазон [10;90])	Дискретно, кожні 10 с (синхронно з циклом FLC)
Криптографічний профіль (Security Profile)	Мапінг значення Security Score, записується у security_policy.json та зчитується сервісом обробки запитів main.py	Дискретні лінгвістичні значення (Safe, Survival, Balanced, Max)	Переоцінка агентом – кожні 10 с; застосування – асинхронно під час кожного HTTP-запиту
Інтегральний критерій якості (J)	Математична модель оцінки компромісу в asm_agent.py (лінійна комбінація нормованих витрат енергії, затримки та рівня безпеки)	Нормалізована безрозмірна величина	Дискретно, кожні 10 с (за результатами агрегації логів ingest.log)

Підсистема ASM-Agent періодично зчитує файл, у якому описаний стан загрози, обробляє Threat Level та разом із рівнем заряду батареї (Battery Level) передає його як вхід у нечіткий логічний контролер (FLC), що реалізує логіку енергоорієнтованого вибору профілю. У реальних розгортаннях інтегральний показник Threat Level формують на основі поєднання виходів систем виявлення вторгнень, алгоритмів виявлення аномалій та засобів моніторингу якості обслуговування (QoS), які разом відображають як наявність атак, так і їхній вплив на роботу fog-/edge-сервісів. Такий підхід відповідає практиці побудови багатокритеріальних моделей загроз для IoT-, fog- та edge-середовищ, де для прийняття рішень щодо безпеки використовують агреговану оцінку, що поєднує сигнали з IDS, детекторів аномалій і показників продуктивності мережі [79].

У моделі загроз інтегральний показник Threat Level узгоджено з потребою одночасно враховувати класичні властивості безпеки (конфіденційність, цілісність, доступність) та здатність fog-вузла зберігати працездатність під час тривалих атак, що виснажують енергетичні ресурси. Така інтерпретація Threat Level відображає концепцію безпеки, орієнтованої на живучість, у межах якої системи fog/edge-комп’ютингу мають підтримувати прийнятний рівень захисту й якості сервісу внаслідок контрольованої деградації функціональності та енергетично обізнаних протоколів в умовах тривалого впливу загроз [80].

3.1.3 Модель віртуальної батареї та облік навантаження центрального процесора

Модель віртуальної батареї в ASM-агенті описує динаміку заряду батареї як функцію часу та поточного завантаження CPU, що є наслідком обраного криптографічного профілю. Стан заряду батареї оновлюють кожного інтервалу Δt із застосуванням деградації, яка лінійно інтерполюється між мінімальною швидкістю розряду та максимальною, пропорційно до виміряного середнього завантаження CPU за інтервал:

$$B_{k+1} = B_k - \Delta t \cdot \left(r_{min} + (r_{max} - r_{min}) \cdot C_{cpu}\right) \quad (3.1)$$

де r_{min} та r_{max} є константами моделі, які підлягають калібруванню на основі експериментальних вимірювань під час роботи вузла у профілі Max із максимальним криптографічним навантаженням. На апаратному рівні складова r_{min} зумовлена струмами витоку напівпровідникових структур (статичне енергоспоживання транзисторів) та роботою периферійних модулів у режимі очікування. Водночас параметр r_{max} відображає динамічне енергоспоживання під час інтенсивного перемикання логічних вентилів процесора в процесі виконання криптографічних операцій. Фізично ця динамічна складова є пропорційною ємності навантаження транзисторів, квадрату напруги живлення та тактовій частоті ($P_{dyn} \approx C \cdot V^2 \cdot f$).

Використання реальної метрики завантаження процесора (CPU utilization), яку отримують із застосуванням функцій моніторингу (наприклад, psutil.cpu_percent() із нульовим інтервалом блокування), створює можливість пов’язати програмне криптонавантаження в ASM API з фактичним використанням апаратних ресурсів кристала. Це на практиці дає змогу оцінювати відносні енергетичні витрати без обов’язкової інтеграції апаратних датчиків струму в HiL-стенд. Такий підхід, орієнтований на вимірювання часу виконання та завантаження CPU як проксі-величин для енерговитрат, широко застосовують у роботах із моделювання та порівняння енергоефективності протоколів безпеки на fog- та edge-платформах [81].

Моніторинг також охоплює оцінювання середньої затримки оброблення запитів API, яку визначають на основі журналу, де ASM API зберігає для кожного HTTP-запиту часову мітку та час оброблення proc_time_ms. ASM-Agent зчитує журнал у межах заданого в конфігурації вікна часу, обчислює середню затримку та частоту запитів, що забезпечує зв’язок між вибором криптопрофілю, навантаженням fog-вузла та якістю обслуговування.

3.1.4 Інтеграція рівня загроз та адаптивного керування криптопрофілем

Нечіткий контролер (FLC) в ASM-агенті формує оцінку безпеки (Security Score) та криптопрофіль на основі двох вхідних змінних: заряду батареї (Battery Level) та Threat Level. Для Battery Level визначено три лінгвістичні терми (LOW, MED та HIGH) із трапецієподібними та трикутними функціями належності, що охоплюють усі діапазони заряду від низьких до високих значень, а для Threat Level – терми LOW, MED, та HIGH із перекриттями в проміжній області.

База правил FLC реалізує продукційні залежності виду «якщо Battery Level is LOW і Threat Level is HIGH, то Security Profile is SURVIVAL» з подальшою агрегацією через оператори min/max та дефазифікацією методом зваженого середнього для моделі Такаґі-Сугено нульового порядку (zero-order). Схеми нечіткого керування нульового порядку демонструють достатню інтерпретованість і придатність для вбудованих систем із обмеженими ресурсами.

На основі отриманого значення Security Score FLC виконує порогову класифікацію для вибору профілю: за значень вище 80 активують профіль максимального захисту (Max), у діапазоні від 50 до 80 – збалансований (Balanced), від 20 до 50 – режим Survival, нижче 20 – безпечне вимкнення (Safe shutdown). Такий пороговий механізм запобігає ефекту брязкоту (chattering) між режимами і забезпечує стабільність криптопрофілю під час малих флуктуацій Battery Level чи Threat Level, що є критичним для fog-вузлів з обмеженими ресурсами керування. Використання FLC для спільного врахування показників безпеки (ризику, надійності обміну) та енергоефективності узгоджується із сучасними підходами до нечіткого енергоменеджменту й керування ресурсами в IoT- та інших електронних системах, де нечіткі регулятори застосовують для балансування енергоспоживання, продуктивності та пов’язаних із безпекою характеристик у єдиній системі правил [82].

3.1.5 Застосування профілю

Сервіс ASM API реалізовано на базі FastAPI та REST-інтерфейсу, який приймає HTTP-запити від IoT-пристроїв, емулюючи оброблення телеметрії або команд керування. Під час кожного запиту API зчитує поточний профіль безпеки зі спільного файлу security_policy.json (через функцію get_current_profile()), після чого викликає simulate_crypto_work(profile) для емулювання криптонавантаження над буфером заданого розміру.

Реалізовано чотири рівні навантаження: профіль maxsecurity виконує великий обсяг ітерацій гешування SHA-512, balanced – менший обсяг ітерацій SHA-256, survival – ще менший обсяг ітерацій SHA-1, safeshutdown – мінімальний обсяг ітерацій MD5. Такі налаштування формують можливість моделювати зростання вартості криптозахисту в термінах навантаження CPU та затримки оброблення під час переходу до ресурсомісткіших алгоритмів.

Після завершення емульованого криптонавантаження ASM API вимірює час оброблення proc_time_ms та записує у файл журналу структуру з полями ts, deviceid, profile, proctimems та іншими допоміжними даними. Ці дані використовує ASM-Agent для подальшого обчислення середньої затримки в інтервалі керування та інтегрального критерію якості J, який відображає компроміс між latency, економією енергії та рівнем безпеки профілю. У результаті ASM-Agent та ASM API формують замкнений контур, де рішення FLC безпосередньо впливають на навантаження CPU, а отже – на швидкість розряду віртуальної батареї та подальший вибір криптопрофілю.

3.1.6 Контрольована деградація функціональності у реалізованому контурі

Survival-концепція в ASM-фреймворку полягає в цілеспрямованому зниженні рівня криптонавантаження за умови критичного падіння Battery Level з одночасним збереженням мінімально прийнятних властивостей конфіденційності та цілісності. У нечіткій базі правил за низького заряду батареї (LOW) навіть за високих значень

Threat Level (HIGH) переважним стає вибір профілю Survival або, в разі подальшого погіршення стану, Safe/Safe shutdown, що відображає пріоритет живучості fog-вузла над максимізацією рівня захисту.

У статичному описі профілів survival-режим асоціюють із використанням відносно легких механізмів автентифікованого шифрування, таких як ChaCha20-Poly1305 у поєднанні з простими засобами контролю цілісності, тоді як профіль Safe/Safe shutdown може спиратися на ще менш затратні в обчислювальному сенсі алгоритми, орієнтовані на мінімальний контроль цілісності даних. Такий вибір відповідає сучасним рекомендаціям щодо полегшеної криптографії (lightweight cryptography) для енергообмежених IoT-пристроїв та fog-вузлів, де ChaCha20-Poly1305 та інші легкі схеми демонструють істотне зменшення часу виконання й енергоспоживання порівняно з традиційними криптопротоколами [83].

3.1.7 Логування та відтворюваність експериментів

ASM-Agent здійснює детальне логування динаміки стану вузла в CSV-файл, де для кожного кроку керування записують часові мітки, поточний заряд батареї (Battery Level), Threat Level, значення Security Score, обраний криптопрофіль, інтегральний критерій якості J, середню затримку оброблення запитів за інтервал, реальний відсоток завантаження CPU та температуру. Такий рівень деталізації забезпечує можливість виконувати статистичний аналіз поведінки системи, порівнювати різні стратегії керування (Adaptive Full Battery, Adaptive Low Battery, Fixed Max, Fixed Balanced) та аналізувати вплив параметрів FLC і ваг цільового критерію.

Використання Docker-оркестрації та конфігурацій JSON-файлів дає змогу відтворювати експерименти з різними початковими рівнями заряду батареї, шаблонами змін Threat Level та наборами ваг w_L, w_P, w_S, що є критичною вимогою наукової валідності запропонованого методу. Логічна ізоляція сервісів у контейнерах спрощує перенесення ASM-фреймворку на інші fog-платформи, забезпечуючи однакову поведінку FLC-контролера та механізмів профілювання безпеки на різних апаратних конфігураціях.

3.2 Нечіткий контролер для вибору криптопрофілю

Нечіткий логічний контролер (FLC, Fuzzy Logic Controller) у складі фреймворку адаптивного менеджера безпеки (ASM) виконує функції енергоорієнтованого адаптивного керування криптографічним захистом fog-вузла, поєднуючи інформацію про стан енергетичних ресурсів та актуальний рівень кіберзагроз у єдину числову оцінку безпеки (Security Score). У сучасних роботах з енергоорієнтованого керування та безпеки в IoT-системах нечітку логіку застосовують для моделювання експертних правил ухвалення рішень в умовах невизначеності параметрів середовища, змінного навантаження та неповної інформації про ризики або загрози. Доведено, що такі нечіткі (fuzzy) моделі забезпечують більш плавну й інтерпретовану адаптацію політик (зокрема енергоспоживання та безпеки) порівняно з жорсткими пороговими процедурами, де малі зміни вхідних параметрів здатні спричиняти різкі зміни рішень [84].

3.2.1 Місце нечіткого контролера в архітектурі фреймворку

У фреймворку ASM fog-вузол на базі ARM SoC функціонує як програмно-апаратна платформа, на якій розгорнуто контейнеризовані сервіси, зокрема ASM API та модуль ASM-Agent, що реалізує нечіткий логічний контролер.

ASM-Agent періодично, із заданим інтервалом, зчитує оновлені значення стану заряду батареї fog-вузла (Battery Level) та показник загроз (Threat Level), обчислює поточний Security Score та на основі цього оновлює файл, який визначає активний криптопрофіль для ASM API. Така організація забезпечує слабке зв’язування між рівнем моніторингу ресурсів і рівнем криптографічних протоколів: FLC інкапсулює експертні правила адаптації, тоді як прикладні сервіси лише інтерпретують обраний профіль і підключають відповідні примітиви (AES-GCM, ChaCha20, HMAC тощо). Структурну схему контролера у складі фреймворку наведено на рис. 3.3.

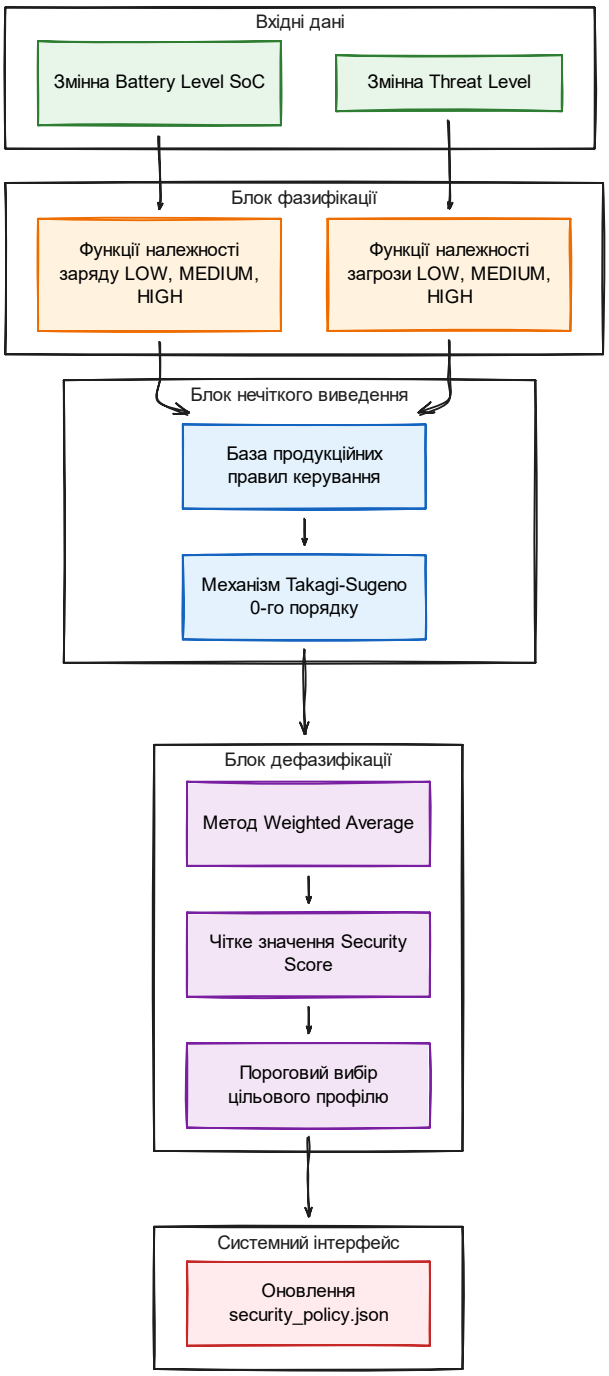

Рисунок 3.3 – Структурна схема нечіткого контролера FLC у складі ASM-фреймворку

FLC в ASM-Agent виконує роль механізму прийняття рішень, узагальнюючи експертні правила типу «якщо заряд батареї низький та загроза висока, зменшити криптографічне навантаження» у формалізовану базу нечітких правил. У роботах, присвячених застосуванню нечіткої логіки в промислових та IoT-системах, показано, що fuzzy-підходи дають змогу явно враховувати кілька конкурувальних критеріїв –

зокрема безпеку/ризик, продуктивність (QoS) та енергетичну ефективність – у єдиній системі правил, зберігаючи інтерпретованість ухвалених рішень для адміністратора [85]. У запропонованому ASM-Agent нечіткий регулятор (FLC) періодично обчислює узагальнений показник Security Score на основі контекстних і енергетичних параметрів та обирає відповідний криптографічний профіль, забезпечуючи адаптивну поведінку fog-вузла з дотриманням вимог конфіденційності, цілісності та доступності в умовах обмежених енергоресурсів.

3.2.2 Вхідні змінні нечіткого контролера

У реалізованому в ASM-Agent варіанті нечіткого регулятора використовують дві основні вхідні змінні – Battery Level та Threat Level, що забезпечує достатню чутливість до енергетичного стану fog-вузла та динаміки атак за помірної складності бази правил. У спеціалізованих роботах з енергетичного менеджменту та нечіткого керування для IoT зазначають, що збільшення кількості вхідних змінних підвищує гнучкість, але зумовлює зростання кількості правил і ускладнює налаштування та інтерпретацію FLC, тому доцільно починати з мінімального набору ключових параметрів (зокрема, стану енергоресурсів та показників ризику) [86]. Потенційне розширення FLC за рахунок додаткових змінних, наприклад поточному завантаженню CPU, розглядають як подальший етап розвитку моделі, але не використовують у поточній реалізації, щоб зберегти компактність і керованість бази правил.

3.2.3 Лінгвістичні змінні та функції належності

Для Battery Level вводять лінгвістичну змінну із трьома термами: LOW, MEDIUM, HIGH, що відображають якісні стани заряду fog-вузла. У реалізованій конфігурації ASM-Agent задають такі діапазони: LOW відповідає зонам низького заряду (орієнтовно нижче ніж 20%), MEDIUM відповідає проміжній зоні (приблизно 20–60%), а HIGH – зоні високого заряду (понад 60%). Для термів LOW і HIGH використовують трапецоїдні функції належності (trapmf), що дають змогу задати

плато високої належності за дуже низьких і дуже високих значень заряду батареї, тоді як для MEDIUM застосовують трикутну функцію належності (trimf), сконцентровану в середньому діапазоні. Поєднання trapmf/trimf забезпечує плавні переходи між станами без різких стрибків вихідної величини в разі малих варіацій входу. Графіки функцій належності показано на рис. 3.4.

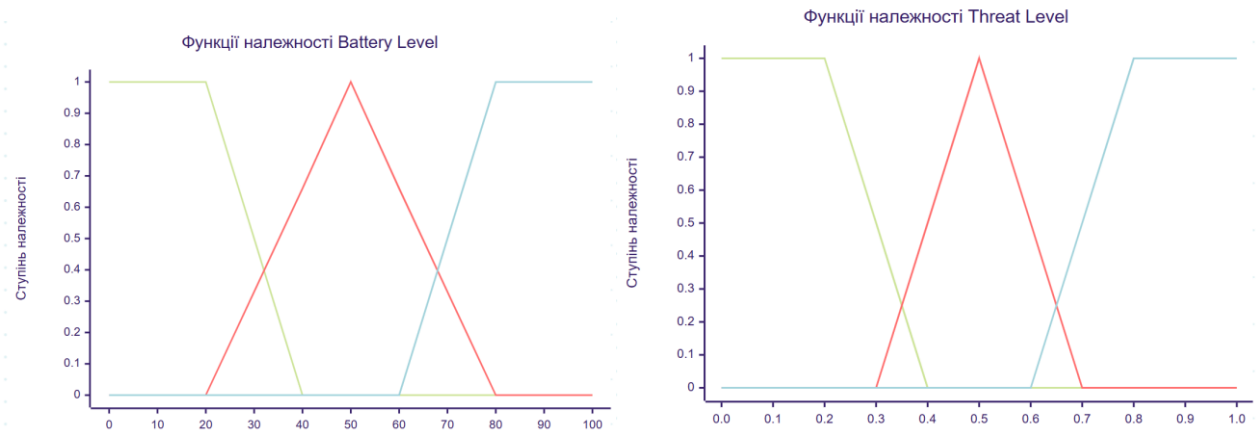

Рисунок 3.4 – Графіки функцій належності для Battery Level та Threat Level

Для Threat Level також визначають лінгвістичну змінну з трьома термами: LOW, MEDIUM, HIGH, кожен з яких відображає якісно різні сценарії кіберзагроз. Терму LOW відповідають значення Threat Level поблизу 0, які інтерпретують як стабільний стан без суттєвих аномалій трафіку, MEDIUM описує проміжний діапазон, де фіксують підвищене навантаження, окремі підозрілі інциденти чи помірні відхилення поведінки, HIGH охоплює значення поблизу 1, що пов'язують із явними ознаками атак (наприклад, розподілені атаки відмови в обслуговуванні або масові спроби несанкціонованого доступу). Для Threat Level, аналогічно до Battery Level, використовують поєднання трапецоїдних і трикутних функцій належності з перекриттям, що дає змогу FLC реагувати не лише на різкі стрибки показника загрози, а й на поступове накопичення ризиків.

Подання критеріїв безпеки у вигляді лінгвістичних змінних LOW/MEDIUM/HIGH із нечіткими межами формує можливість інтегрувати як кількісні, так і якісні ознаки загроз та ресурсних обмежень, а також спростити формулювання експертних правил. Для ASM-фреймворку це означає, що менеджер безпеки здатний інтерпретувати функціонування FLC через зрозумілі терми Battery

Level LOW/HIGH і Threat Level HIGH/LOW, не вдаючись до складної математики, але система продовжує виконувати формалізований нечіткий висновок.

3.2.4 Вихідна змінна рівня безпеки

Вихідну змінну нечіткого контролера в ASM-фреймворку визначають як Security Score – нормалізовану числову оцінку бажаного рівня криптографічного захисту fog-вузла в діапазоні від 0 до 100.

У реалізації Такаґі-Сугено нульового порядку (zero-order T-S) вихід контролера задають як одну з чотирьох фіксованих констант Security Score, що відповідають криптографічним профілям: OUT_SAFE = 10 для профілю Safe (або Safe-shutdown), OUT_SURVIVAL = 30 для профілю Survival, OUT_BALANCED = 60 для профілю Balanced та OUT_MAX = 90 для профілю Max. Така параметризація відповідає класичному підходу до побудови zero-order T-S-систем, у яких кожне нечітке правило породжує сталий рівень керувального впливу. Приклад бази правил наведено в табл. 3.2.

Таблиця 3.2 – Приклад бази правил FLC

	Рівень заряду батареї		
Рівень загрози	Низький	Середній	Високий
Низький	«SAFE»	«SURVIVAL»	«SURVIVAL»
Середній	«BALANCED»	«BALANCED»	«BALANCED»
Високий	«BALANCED»	«MAX»	«MAX»

Константи OUT_SAFE, OUT_SURVIVAL, OUT_BALANCED та OUT_MAX входять у праві частини нечітких правил як фіксовані значення вихідної функції, що активуються залежно від ступеня виконання відповідних умов щодо Battery Level і Threat Level. Така структура відповідає zero-order реалізації нечіткого (Takagi-Sugeno-подібного) контролера, у якій кожне правило генерує сталий керувальний рівень, а фактичний вихід формують шляхом агрегації внесків активних правил. Така побудова дає змогу однозначно пов’язати кожну комбінацію лінгвістичних термів із певним

класом криптографічного профілю, водночас забезпечуючи плавне змішування результатів кількох правил у зонах перекриття функцій належності, що є типовим для нечітких систем керування.

У літературі з нечітких систем керування для енергоорієнтованих застосувань доведено, що представлення виходу у вигляді невеликої кількості дискретних рівнів із фіксованими значеннями спрощує аналіз компромісів між енергоспоживанням, якістю сервісу та затримкою, оскільки кожному рівню відповідає наперед визначений режим роботи системи. У задачах енергоефективного криптографічного оброблення даних аналогічний підхід дає змогу поставити у відповідність кожному дискретному рівню керування конкретний набір криптографічних алгоритмів або параметрів їхнього налаштування, що створює можливість явно описати компроміси між енергоспоживанням, затримкою та рівнем захисту в ресурсно-обмежених вбудованих системах [87].

3.2.5 База правил нечіткого контролера

Базу правил нечіткого контролера ASM-Agent побудовано з урахуванням принципу енергоорієнтованої адаптації: в умовах низького заряду батареї (Battery Level) система допускає зниження криптографічного навантаження внаслідок переходу до легших профілів, тоді як в умовах високих загроз (Threat Level) FLC намагається утримати профілі з вищим рівнем захисту, навіть ціною збільшення енергоспоживання. На концептуальному рівні база правил охоплює всі дев’ять комбінацій Battery ∈ {LOW, MEDIUM, HIGH} та Threat ∈ {LOW, MEDIUM, HIGH}, для кожної з яких задають відповідний вихідний рівень Security Score через константи OUT_SAFE, OUT_SURVIVAL, OUT_BALANCED або OUT_MAX. Наприклад, комбінація Battery LOW і Threat HIGH зумовлює вихід, близький до OUT_SURVIVAL; Battery MEDIUM і Threat HIGH – до OUT_BALANCED; Battery HIGH і Threat HIGH – до OUT_MAX. Формально нечіткий висновок у схемі Такаґі-Сугено 0-го порядку реалізують за класичною процедурою: для кожного правила i обчислюють ступінь активації w_i як функцію значень належності вхідних змінних до відповідних

лінгвістичних термів (зазвичай через операції мінімуму або добутку), після чого вихід правила z_i дорівнює відповідній константі (OUT_SAFE, OUT_SURVIVAL, OUT_BALANCED або OUT_MAX). Після активації всіх правил вихідний Security Score обчислюють методом зваженого середнього (weighted average):

$$SecurityScore = \frac{\sum_{i=1}^{N} w_i \cdot z_i}{\sum_{i=1}^{N} w_i}, \tag{3.2}$$

де N – кількість правил у базі, w_i – ступінь активації правила i, z_i – фіксоване значення виходу цього правила. Така схема забезпечує гладку залежність Security Score від Battery Level і Threat Level, що особливо важливо для енергообмежених платформ, де різкі стрибки профілю безпеки здатні спричинити небажані піки навантаження на CPU та мережу.

У роботах із нечіткого керування для технічних систем відзначено, що підхід Такаґі-Сугено з константними або лінійними наслідками та дефазифікацією у вигляді зваженого середнього характеризується порівняно низькою обчислювальною складністю і придатністю до реалізації на вбудованих платформах реального часу [88]. У контексті ASM-фреймворку це створює можливість реалізувати FLC як полегшений компонент (наприклад, у Python або C/C++) усередині Docker-контейнера на ARM-базованому fog-вузлі без помітного впливу на продуктивність прикладних сервісів, за умови невеликої кількості правил та вхідних змінних.

3.2.6 Механізм запобігання брязкоту та порогова дискретизація рівня безпеки

Навіть за умови гладкої залежності Security Score від вхідних змінних, невеликі коливання Battery Level або Threat Level поблизу меж між лінгвістичними термами здатні спричиняти часті перемикання криптопрофілів, що є небажаним з огляду на стабільність QoS та накладні витрати на ініціалізацію криптографічних механізмів. Для зменшення ефекту брязкоту (chattering) в ASM-фреймворку використано

порогову дискретизацію виходу, коли неперервний Security Score відображають в один із чотирьох дискретних профілів за фіксованими діапазонами. Така схема дає змогу стабілізувати вибір профілю в умовах невеликих флуктуацій виходу, зводячи частоту перемикань до прийнятного рівня без втрати адаптивності на макрорівні.

Для ASM-фреймворку є можливість розширення цієї схеми за рахунок введення гістерезису (різні пороги для переходів між профілями «вгору» та «вниз») і додаткового усереднення Battery Level та Threat Level на ковзному часовому вікні, що формує можливість додатково покращити стійкість профілю до швидких короткочасних змін. Порогові діапазони наведено у табл. 3.3.

Таблиця 3.3 – Порогові діапазони для відповідності значень Security Score криптопрофілям

Діапазон Security Score	Рекомендований криптопрофіль	Призначення та роль у механізмі запобігання брязкоту (chattering)
< 20	Safe	Активується за вкрай низьких значень оцінки. Діапазон відтинає незначний фоновий шум і не дає змогу системі випадково активувати криптографічне навантаження.
20 – 49	Survival	Широка буферна зона (30 одиниць), що забезпечує стабільне утримання енергоощадного режиму та запобігає стрибкоподібному поверненню до профілю Safe на межі значень.
50 – 79	Balanced	Основний робочий інтервал. Гарантує стабільне утримання збалансованого захисту за помірних флуктуаціях інтенсивності трафіку та заряду батареї.
≥ 80	Max	Верхній поріг активації. Система перемикається у найважчий обчислювальний режим лише за високої впевненості у загрозі, що повністю виключає хибні спрацювання профілю Max.

3.2.7 Узгодження нечіткого контролера з інтегральним критерієм J

Функціонування FLC в ASM-фреймворку узгоджено з інтегральним критерієм якості J. В ASM-фреймворку FLC не мінімізує J аналітично, однак його базу правил та функції належності налаштовано так, щоб під час експериментальної перевірки на

HiL-стенді забезпечувати менші значення J у сценаріях Full Battery та Low Battery порівняно зі статичними стратегіями Fixed Max та Fixed Balanced. Внаслідок цього результати роботи FLC піддаються не лише якісному, а й кількісному оцінюванню за метриками latency/energy/security.

3.3 Криптографічні профілі та політики перемикання

Криптографічний профіль fog-вузла описують як формалізовану комбінацію алгоритмів шифрування, автентифікації та контрольних механізмів, що визначає рівень безпеки, обчислювальне навантаження та енерговитрати під час оброблення та маршрутизації трафіку на енергообмеженому ARM-базованому вузлі. У реалізованому ASM-фреймворку криптографічний профіль P_i можна подати у вигляді впорядкованого набору параметрів:

$$P_i = \{A_{enc}, A_{auth}, M_{trans}, C_{check}\}, \tag{3.3}$$

де A_{enc} описує алгоритм шифрування (AES-GCM, ChaCha20-Poly1305 тощо), A_{auth} – механізм криптографічної автентифікації (HMAC-SHA-256, HMAC-SHA-512), M_{trans} – механізм забезпечення цілісності на транспортному рівні (автентифіковане шифрування або окремий код автентифікації повідомлення (MAC)), а C_{check} – додаткові контрольні процедури (CRC32, легкі контрольні суми). Для кожного профілю у межах процедури обчислення інтегрального критерію якості J відповідно задають скалярну оцінку безпеки (Security Score), що відображає ефективність профілю щодо конфіденційності, цілісності і стійкості до атак в умовах фіксованого набору загроз.

Профіль Safe (або Safe shutdown) у фреймворку ASM призначений для роботи fog-вузла в режимі мінімального енергоспоживання та обчислювального навантаження, коли можливості системи щодо підтримки повноцінних криптографічних механізмів є суттєво обмеженими через низький рівень заряду батареї або критичне навантаження процесора. У реалізації crypto_profiles.py профіль

Safe моделюють через мінімалістичний криптографічний стек: функція apply_security_overhead застосовує легке гешування (MD5 у simulate_crypto_work в main.py) для емулювання навантаження на CPU та контрольну суму CRC32 для перевірки цілісності корисного навантаження, без використання повноцінного алгоритму автентифікованого шифрування. У контексті інтегрального критерію якості J профіль Safe забезпечує найнижче значення Security Score (10), але різко зменшує компонент P_{norm} (EnergyDelta) за рахунок мінімальної кількості ітерацій у simulate_crypto_work (1000 ітерацій MD5-гешування для типового розміру буфера). Подібні профілі відповідають концепціям контрольованої деградації функціональності, де в умовах зниження доступної енергетичної потужності система зберігає лише базові механізми контролю цілісності і спрощене шифрування або анонімізацію даних, але не переходить у повністю відкритий режим передавання. Опис профілів наведено в табл. 3.4.

Таблиця 3.4 – Опис профілів

Профіль безпеки	**Security Score**	**Застосовані криптографічні алгоритми**	**Якісний рівень навантаження на CPU**
«Safe»	10	Відсутність шифрування або мінімальний контроль цілісності (клас CRC32)	Мінімальний
«Survival»	30	Емульоване навантаження легких криптопримітивів: ChaCha20-Poly1305 зі спрощеним контролем цілісності CRC32	Низький
«Balanced»	60	Емульоване навантаження середньої складності: AES-128-GCM із MAC середньої складності HMAC-SHA-256	Середній
«Max»	90	Емульоване навантаження підвищеної складності: AES-256-GCM із MAC підвищеної складності HMAC-SHA-512	Високий

Профіль Survival розроблено як полегшений, але криптографічно прийнятний режим, орієнтований на забезпечення автентифікованого шифрування в умовах істотних обмежень енергетичного бюджету. У модулі crypto_profiles.py цей профіль

застосовує алгоритм ChaCha20-Poly1305 як основний механізм автентифікованого шифрування, що поєднує конфіденційність і цілісність у єдиному примітиві, а також використовує CRC32 як додатковий обчислювально дешевий контроль під час оброблення трафіку з невеликою кількістю обчислень. В емуляції CPU-навантаження у main.py Survival відповідає режиму SHA-1 із середнім числом ітерацій (6000), що розташовується між Safe (MD5, 1000 ітерацій) та Balanced (SHA-256, 15000 ітерацій) за обчислювальною складністю. З апаратної точки зору, ChaCha20-Poly1305 є потоковим програмно-оптимізованим алгоритмом, який ефективно виконується на базових конвеєрах арифметико-логічного пристрою (АЛП). Це робить його незамінним для бюджетних IoT-вузлів, які не мають виділеного апаратного криптографічного співпроцесора, або в ситуаціях, коли такий співпроцесор примусово вимкнено (переведено в режим сну) задля максимального енергозбереження. ChaCha20-Poly1305 можна розглядати як одну з рекомендованих полегшених схем автентифікованого шифрування з приєднаними даними (AEAD), яка демонструє сприятливий компроміс між енергоспоживанням, затримками та стійкістю до сучасних криптоаналітичних атак на платформах з обмеженим апаратним прискоренням стандарту AES. У профілі Survival ця властивість працює як основа для гарантування мінімально прийнятного рівня безпеки в умовах підвищеного ризику виснаження батареї.

Базовий профіль Balanced є рекомендованим режимом для типових умов роботи fog-вузла з невисоким або середнім рівнем загроз і достатнім запасом енергоресурсів. Реалізація в crypto_profiles.py задає для Balanced використання AES-128-GCM як основного алгоритму автентифікованого шифрування, доповненого HMAC-SHA-256, що формує можливість забезпечити високий рівень конфіденційності й цілісності за умови помірного навантаження на процесор. У модулі main.py профіль Balanced моделюють через SHA-256 із 15000 ітерацій у функції simulate_crypto_work, що відповідає середньому рівню енергоспоживання та затримок серед наявних профілів. Цей режим шифрування забезпечує прийнятний баланс між безпекою й витратами енергії. Висока енергоефективність алгоритму AES зумовлена тим, що сучасні системи на кристалі (SoC) архітектури ARM містять спеціалізовані апаратні

криптографічні розширення (наприклад, ARMv8 Cryptography Extensions). Наявність таких розширень дає змогу виконувати математичні операції шифрування безпосередньо на рівні виділених апаратних регістрів майже без залучення основного конвеєра АЛП, що істотно знижує динамічне енергоспоживання мікросхеми. Навіть без апаратної підтримки цей режим залишається конкурентним щодо альтернативних AEAD-схем. У моделі Security Score профіль Balanced асоціюють із фіксованим показником 60, що відображає його роль як компромісного рішення між безпекою та ресурсними витратами.

Профіль Max (або maxsecurity) орієнтований на сценарії з підвищеним рівнем загроз, де пріоритет надають максимальному рівню криптографічного захисту навіть ціною зростання енерговитрат і затримок. У crypto_profiles.py цей профіль реалізує використання AES-256-GCM як алгоритму автентифікованого шифрування з 256-бітним ключем, доповненого HMAC-SHA-512 як механізмом автентифікації, що посилює стійкість до колізій і створює більший запас криптостійкості. У модулі main.py обчислювальне навантаження Max моделюють із застосуванням SHA-512 із 25000 ітерацій у simulate_crypto_work, що формує максимальний час оброблення запитів серед усіх профілів. У моделі Security Score профілю Max відповідає значення 90, що відображає його статус як профілю з максимальним рівнем безпеки в межах підтримуваних режимів. У нечіткій моделі керування безпекою ASM-фреймворку кожному профілю ставлять у відповідність фіксоване значення Security Score: Safe → 10, Survival → 30, Balanced → 60, Max → 90. У класі FuzzySecurityController ці значення реалізовані як константи OUT_SAFE, OUT_SURVIVAL, OUT_BALANCED, OUT_MAX, що використано як виходи правил Такаґі-Сугено нульового порядку, де кожне правило має вигляд:

$$\text{Якщо } x \text{ це } A \text{ тоді } z = C, \tag{3.4}$$

де A – нечітка множина вхідних змінних, а C – одна з констант OUT, що відповідає конкретному профілю. Агрегований вихід FLC обчислюють як зважене середнє:

$$SecurityScore = \frac{\sum w_i \cdot C_i}{\sum w_i}, \quad (3.5)$$

де w_i – ступені спрацьовування правил, що відповідають різним комбінаціям зон Battery Level та Threat Level. Нормоване значення Security Score далі входить до інтегрального критерію якості J, де компонент безпеки враховують у вигляді S_{norm} із ваговим коефіцієнтом w_S. Приклади політик перемикання наведено в табл. 3.5.

Таблиця 3.5 – Приклади політик перемикання

Зона Battery Level	**Зона Threat Level**	**Нечіткий вихід FLC (Security Score)**	**Рекомендований криптографічний профіль**
НИЗЬКА	НИЗЬКА	10	«Safe»
НИЗЬКА	СЕРЕДНЯ	60	«Balanced»
НИЗЬКА	ВИСОКА	60	«Balanced»
СЕРЕДНЯ	НИЗЬКА	30	«Survival»
СЕРЕДНЯ	СЕРЕДНЯ	60	«Balanced»
СЕРЕДНЯ	ВИСОКА	90	«Max»
ВИСОКА	НИЗЬКА	30	«Survival»
ВИСОКА	СЕРЕДНЯ	60	«Balanced»
ВИСОКА	ВИСОКА	90	«Max»

Фіксовані значення Security Score також використовують для порогової дискретизації виходу FLC у конкретний профіль, до того ж регіони для Safe, Survival, Balanced та Max вибрано так, щоб зменшити ризик брязкоту в умовах невеликих коливань вхідних параметрів. Правила дискретизації визначають, що Safe обирають за умови $Score < 20$, Survival – у діапазоні $[20,50)$, Balanced – за умови $[50,80)$, Max – якщо $Score \geq 80$. Таким чином криптографічні профілі інтегрують не лише на рівні реалізації бібліотек, а й у математичну модель адаптації, де показник безпеки (Security Score) входить до складу інтегрального критерію якості J як керована величина, що балансує з нормованими показниками затримки (latency) та стану заряду батареї (Battery Level). Подібні багатокритеріальні постановки, у яких одночасно враховують енергоспоживання, затримку та показники якості/надійності сервісу, є типовими для

задач керування в автономних системах Інтернету речей. У літературі з енергообізнаної безпеки (energy-aware security) для енергообмежених IoT-систем відзначено потребу спільного керування параметрами безпеки та енергоспоживанням, де рівень шифрування й автентифікації описують як адаптивну змінну, що підлаштовується під поточний енергетичний статус та вимоги до якості обслуговування. Відповідні оглядові роботи рекомендують включати параметри криптографічних механізмів до цільових критеріїв, щоб явно описувати компроміси між тривалістю життя батарейних вузлів, затримкою та рівнем захисту [89].

Модель Battery Level в ASM-фреймворку базують на поданні стану заряду батареї fog-вузла у відсотках, з подальшим розбиттям цього діапазону на зони LOW, MEDIUM, HIGH, які використовують як лінгвістичні терми у функціях належності FLC. Функції fuzzify_battery в asm_agent.py реалізують трапецієподібні та трикутні функції належності: зону LOW описують як trapmf з опорою в нижньому інтервалі, MED як trimf з максимумом у середньому діапазоні (приблизно 50%), HIGH як trapmf зі зростанням від $\approx 60\%$ до 100%. Для потреб керування профілями вводять поняття критичного заряду батареї, що відповідає зоні Battery LOW, у якій система має переходити до енергоощадних профілів Survival або Safe.

У базі правил FLC пороги Battery Level використовують як умови, що обмежують можливість обрання криптографічно складних профілів в умовах низького заряду батареї, що реалізує енергоорієнтовану політику керування рівнем захисту.

В умовах Battery LOW, навіть за високих значень Threat Level, активують правила, які віддають перевагу профілям Survival або Safe, що зменшує EnergyDelta та CPU-навантаження і тим самим уповільнює розряджання батареї.

За умов Battery MEDIUM або HIGH FLC формує можливість застосовувати профілі Balanced і Max залежно від значення Threat Level, забезпечуючи вищий Security Score у сценаріях, де ризик компрометації даних чи порушення доступності домінує над енергетичними ризиками. Це відповідає рекомендаціям оглядових робіт з energy-aware security та полегшеної криптографії, де наголошено на потребі

врахування залишкового ресурсу батареї та пріоритизації трафіку під час вибору сили криптографічного захисту [90].

Змінна Threat Level у нечіткій моделі ASM відображає узагальнену оцінку рівня загроз для fog-вузла, яка здатна базуватися на показниках інтенсивності запитів (RPS), результатах систем IDS, характері сплесків навантаження, що нагадують DDoS-атаку, та інших сигналах про аномальну активність. У функції fuzzify_threat в asm_agent.py Threat Level нормують до діапазону реальних значень, після чого описують як композицію зон LOW, MEDIUM і HIGH, де LOW відповідає нормальній фоновій активності, MEDIUM – підвищеному навантаженню або окремим підозрілим подіям, HIGH – стабільно високому рівню підозрілої активності, що свідчить про DDoS-подібні атаки або масові сканування. У роботах із виявлення DDoS-атак у fog-орієнтованих IoT-системах подібні інтегральні показники загрози використовують як вхідні параметри для систем класифікації, що в режимі реального часу адаптують політики фільтрації, пріоритизації та криптографічного захисту. У ASM-фреймворку зона Threat HIGH є тригером для правил, що віддають перевагу профілям Balanced або Max, за умови, що Battery Level не перебуває у критичній зоні LOW, де енергетичні обмеження домінують над вимогами до безпеки.

Політику перемикання між криптографічними профілями в ASM-фреймворку задають композицією нечітких правил над змінними Battery Level та Threat Level, вихідного Security Score та порогової дискретизації цього виходу до одного з чотирьох профілів. Фактична логіка ухвалення рішення має такі узагальнені принципи: за умов HIGH Battery та HIGH Threat система переходить або залишається в профілі Max, за умов MEDIUM Battery та HIGH чи MEDIUM Threat активується профіль Balanced, за умов LOW Battery незалежно від Threat Level застосовують профілі Survival або Safe із блокуванням вибору Max, за умов HIGH Battery та LOW Threat FLC підтримує профіль Balanced, а в разі агресивної політики енергозбереження – навіть Safe. Діаграму переходів між профілями зображено на рис. 3.5.

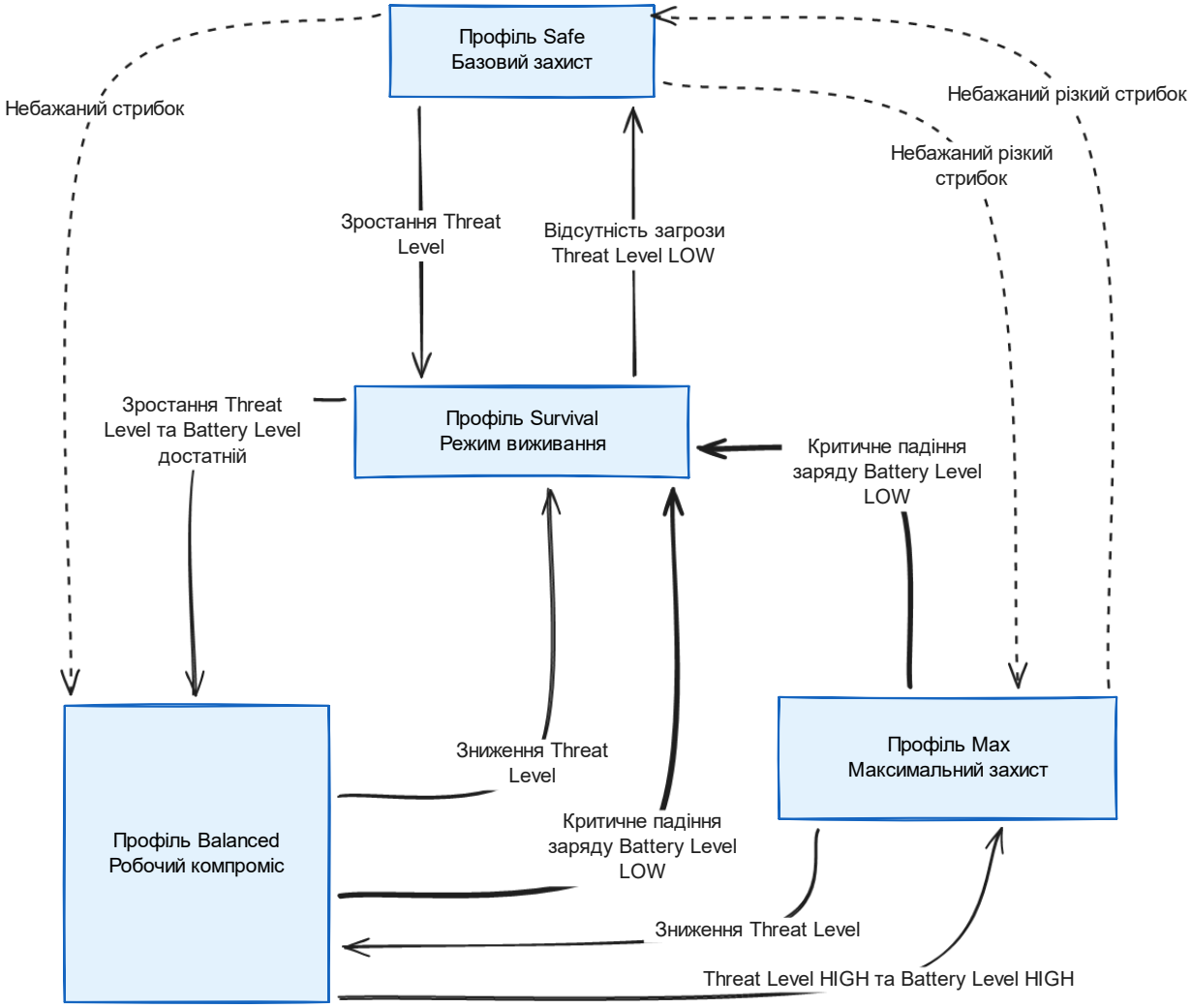

Рисунок 3.5 – Діаграма переходів між профілями

Політика перемикання реалізує енергоорієнтовану адаптацію, оскільки посилений захист профілю Max дозволено лише тоді, коли заряд батареї перебуває в зонах MEDIUM або HIGH і не спостерігається ризику швидкого виснаження акумулятора.

Система гарантує мінімально прийнятний рівень безпеки навіть у режимах Survival та Safe внаслідок обмеження у базі правил FLC і відсутності профілів, які б повністю відключали криптографічний захист.

У профілі Survival використовують ChaCha20-Poly1305, що забезпечує автентифіковане шифрування та криптографічну цілісність корисного навантаження, тоді як CRC32 додатково виявляє помилки на транспортному рівні, розділяючи функції контролю цілісності каналу та криптографічного захисту даних.

У профілі Safe, попри мінімізацію обчислювальних витрат, зберігається щонайменше шифрування або гешування даних разом із контрольними сумами, що перешкоджає переходу fog-вузла до повністю відкритого режиму, несумісного з вимогами до живучості IoT-систем критичного та напівкритичного призначення.

У літературі концепція безпеки, орієнтованої на живучість у fog/edge-сценаріях, ґрунтується на наявності «мінімально допустимого» рівня захисту, який підтримують навіть у разі обмеження енергетичних ресурсів або атак на продуктивність, що узгоджується з інтерпретацією профілю Survival як режиму мінімальної, але прийнятної безпеки [91].

Структура дозволених переходів між профілями обмежується в такий спосіб, щоб уникати небажаних стрибків та осциляцій профілю в умовах незначних флуктуацій Battery Level чи Threat Level. Зокрема, небажаним вважають безпосередній перехід із Safe до Max без проміжного Balanced, якщо Threat Level не демонструє стабільно високих значень, які б обґрунтовували такий стрибок у Security Score і збільшення енерговитрат. За умови падіння заряду батареї нижче від критичної межі в зоні LOW FLC блокує переходи в бік ресурсомісткіших профілів, зберігаючи лише Survival або Safe. Це запобігає швидкому виснаженню акумулятора та формує можливість підтримувати функціональність fog-вузла в режимі керованої деградації функціональності. Хоча в коді політика перемикання формально реалізована через порогову дискретизацію Security Score, її інтерпретація в термінах дозволених і заборонених переходів між профілями є суттєвою для пояснення стійкості та передбачуваності поведінки системи в умовах змінних загроз і ресурсних обмежень. У перспективі доцільно розширити політику внаслідок введення граничної частоти перемикань, гістерезису на межах між профілями та заборони частих коливань типу Balanced ↔ Max без суттєвої зміни Battery Level або Threat Level. Взаємозв’язок криптографічних профілів з інтегральним критерієм якості J виявляється через вплив на три основні складові: нормовану затримку L_{norm}, нормований показник батареї P_{norm} та нормований показник безпеки S_{norm}. В asm_agent.py інтегральний критерій якості J задають у вигляді:

$$J = w_L L_{\text{norm}} + w_P P_{\text{norm}} + w_S(1 - S_{\text{norm}}), \quad (3.6),$$

де w_L, w_P, w_S – вагові коефіцієнти, що відображають пріоритети між затримкою (latency), енергетичною складовою та безпекою. Профіль Max забезпечує найвище значення Security Score, що зменшує третю складовую S_{norm}, але одночасно підвищує L_{norm} та P_{norm} (швидкість розряду) через більші затримки та швидше падіння заряду батареї. Профілі Survival і Safe, навпаки, зменшують L_{norm} і приводять до меншого P_{norm} (повільніше зниження заряду батареї), але знижують Security Score, збільшуючи внесок компоненти безпеки S_{norm} в інтегральний критерій якості J.

Нормування L_{norm}, P_{norm}, S_{norm} виконують за окремими формулами, які враховують діапазони затримки, початковий заряд батареї та шкалу Security Score.

3.4 Алгоритм реалізації та інтеграція у середовище туманних обчислень

Метод енергоорієнтованого адаптивного керування криптографічним захистом fog-вузлів реалізують як контейнеризований ASM-фреймворк на базі ARM SoC, що поєднує FastAPI-службу ASM API, агент адаптивної безпеки ASM-Agent та спільні конфігураційні й журнальні (логові) сховища. Така організація забезпечує ізоляцію компонентів, відтворюваність експериментів та можливість перенесення підходу на інші fog-платформи, сумісні з операційною системою Linux та середовищем Docker.

Fog-вузол побудовано на апаратній платформі типу «система на кристалі» архітектури ARM із 8 ГБ оперативної пам'яті (ОЗП), на якій розгорнуто контейнеризований програмно-апаратний комплекс ASM, що містить сервіс ASM API, агент ASM-Agent та допоміжні JSON-конфігурації й журнальні файли. У розрізі апаратної електроніки, загальний енергетичний бюджет такого fog-вузла жорстко обмежено ємністю фізичного акумулятора. Цей ресурс розподіляють не лише між центральним процесором і ОЗП, а й між енергоємними радіомодулями (Wi-Fi, Bluetooth, LoRa) та апаратними інтерфейсами зв'язку із сенсорними мережами (SPI, I2C, UART). Піковий струм споживання під час інтенсивних обчислень криптографічного процесора здатний спричинити локальне падіння напруги

живлення на системній шині. Таке явище створює ризик апаратного перезавантаження чутливих до живлення радіочастотних трактів або повної втрати зв'язку. Відповідно, ASM функціонує як надбудова над сервісом оброблення IoT-запитів: логіка пристроїв та формат REST-повідомлень залишаються незмінними, натомість зміни зазнає лише обраний криптографічний профіль. За таких умов керування профілями є механізмом підтримки стабільності всієї системи живлення вузла, оскільки воно безпосередньо впливає на динамічне навантаження CPU та інтенсивність віртуального розряджання батареї.

У контейнери ASM API та ASM-Agent передають спільний том із файлами, через які реалізують обмін керувальною інформацією між контуром керування та сервісом оброблення запитів. Така модульна організація узгоджується з типовими fog-архітектурами, де служби безпеки інкапсулюють як окремі сервіси й взаємодіють із застосунками через стандартизовані інтерфейси [92].

3.4.1 Логічна структура реалізації на вузлі туманних обчислень

Програмну реалізацію методу на fog-вузлі з ARM SoC ґрунтують на використанні окремих контейнерів Docker для кожного логічного компонента ASM-фреймворку.

У базовій конфігурації виділяють контейнер ASM API (FastAPI-сервіс, який реалізує REST-інтерфейс для IoT-клієнтів з емульованим криптографічним навантаженням), контейнер ASM-Agent (адаптивний контролер безпеки з нечітким регулятором та енергетичною моделлю батареї) та допоміжні сервіси й томи Docker volume для збереження конфігурацій і логів. Такий підхід логічного розділення компонентів узгоджується з мікросервісно-орієнтованим проєктуванням fog/edge-архітектур, у яких окремі сервіси розгортають в ізольованих контейнерах, завдяки чому їх можливо оновлювати та масштабувати незалежно один від одного [93].

Контейнер ASM API виконує роль фронтенд-шлюзу для IoT-пристроїв і обробляє HTTP-запити ingest, які моделюють типове навантаження сенсорних вузлів або шлюзів, що передають дані до fog-рівня. Усередині контейнера реалізовано

FastAPI-застосунок, який під час кожного запиту визначає актуальний криптографічний профіль, виконує емульоване криптооброблення з різною інтенсивністю завантаження CPU залежно від профілю та веде журнал оброблення у вигляді JSONL-логу. Така конструкція створює можливість зіставляти ухвалені ASM-Agent рішення щодо профілю з реальною затримкою та завантаженням процесора на fog-вузлі, що відповідає підходам до HiL-експериментів для IoT- та fog-інфраструктур. Діаграму розгортання наведено на рис. 3.6.

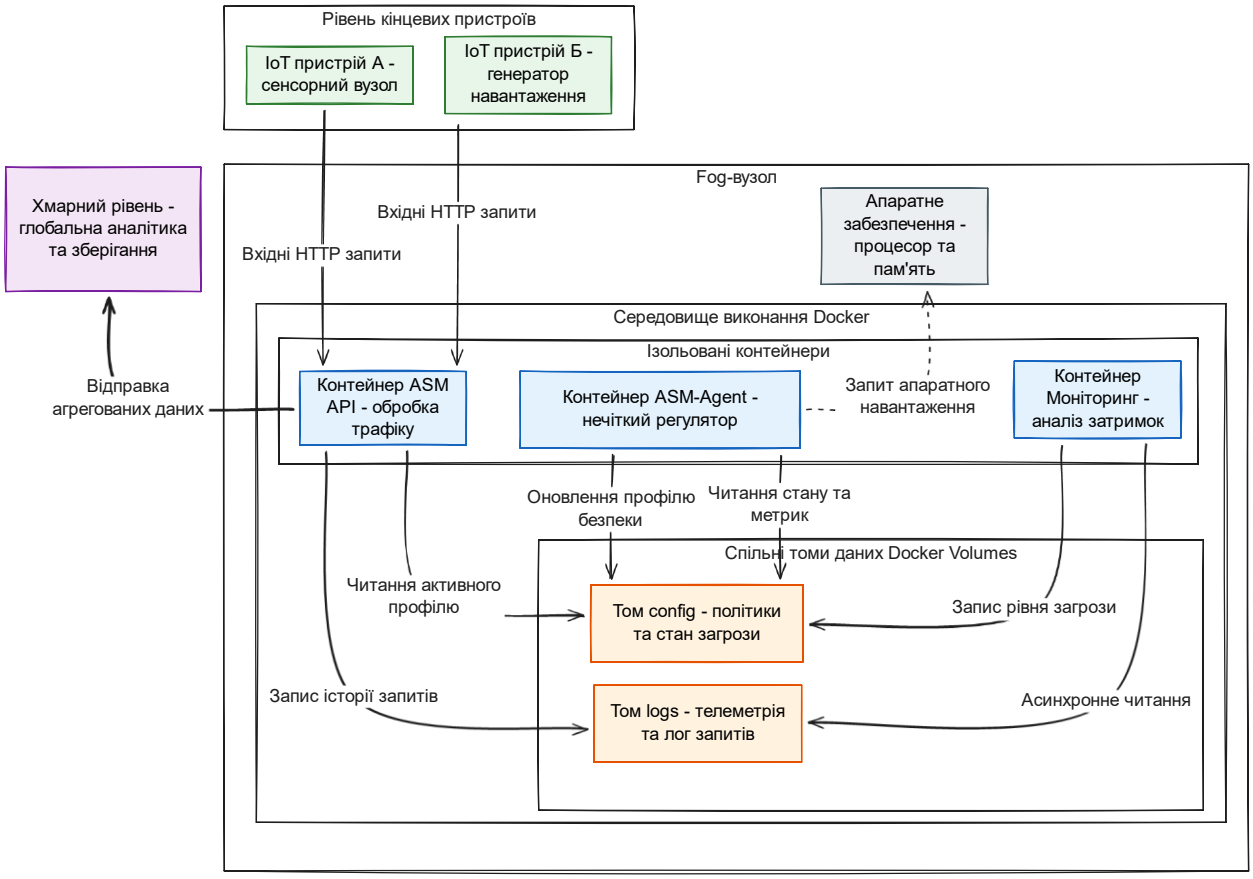

Рисунок 3.6 – Діаграма розгортання ASM-фреймворку на fog-вузлі

Контейнер ASM-Agent реалізує функції FLC-контролера, моніторингу реальних ресурсів fog-вузла (завантаження CPU, температура), ведення енергетичної моделі батареї та обчислення інтегрального критерію якості J для обраного профілю захисту. Цей контейнер не обробляє користувацькі запити безпосередньо, а працює як окремий сервіс керування політикою, що взаємодіє з ASM API через спільні конфігураційні файли та логи в спільному томі. Поділ «шлюз-застосунок / агент адаптивної безпеки» відповідає концепціям рознесення каналів даних та керування в IoT-архітектурах і

спрощує інтеграцію з наявними сервісами, де агент може бути підключений до вже розгорнутого API-шлюзу.

Спільні томи (volumes) Docker призначені для зберігання конфігураційних JSON-файлів (політики безпеки, стану загроз) та логів експериментів, що забезпечує сталість даних між перезапусками контейнерів і дає змогу виконувати серії HiL-експериментів із різними наборами параметрів без втрати історії. Така схема усуває залежність від конкретної версії базової операційної системи (ОС) fog-вузла, оскільки оновлення виконують на рівні контейнерів і конфігурацій.

3.4.2 Алгоритм роботи агенту нечіткого контролера

Основний цикл ASM-Agent реалізований у модулі asm_agent.py як дискретний алгоритм адаптивного керування, що періодично оновлює політику криптографічного захисту залежно від стану енергоресурсів та завантаження fog-вузла. На кожній ітерації агент взаємодіє з конфігураційними файлами, системними сенсорами та нечітким контролером FLC, після чого записує оновлений стан у JSON-файл політики та у CSV-лог експерименту.

На початку ітерації ASM-Agent зчитує поточні параметри політики безпеки із файлу security_policy.json, включно з рівнем заряду батареї (Battery Level), базовим та фінальним профілем (base_profile, final_profile), поточним рівнем загроз (Threat Level) та значенням метрики безпеки Security Score. Файл обробляють через допоміжні функції читання JSON-конфігурацій, які гарантують наявність усіх необхідних полів навіть у разі пошкодження чи відсутності файлу, що підвищує стійкість реалізації.

Поточний рівень загрози додатково зчитують із файлу threat_state.json, який можуть оновлювати зовнішні компоненти, такі як система виявлення вторгнень або керувальні сценарії експерименту. Далі ASM-Agent із застосуванням бібліотек системного моніторингу (psutil) вимірює реальне завантаження CPU та температуру процесора ARM SoC, що формує можливість враховувати поточний стан fog-вузла під час обрання профілю захисту.

Рівень заряду батареї оновлюють згідно з дискретною енергетичною моделлю, яка враховує споживання у станах простою (DRAIN_IDLE) та максимального навантаження (DRAIN_MAX) з урахуванням поточного завантаження CPU.

Оновлені значення Battery Level та (за потреби) показники температури передають до нечіткого регулятора FLC, який реалізує відображення (Battery,Threat) → SecurityProfile з використанням правил типу Такагі-Сугено нульового порядку. FLC формує числовий показник Security Score у діапазоні від Safe до Max, а також обирає один із профілів {Safe, Survival, Balanced, Max} залежно від термів Battery (LOW/MEDIUM/HIGH) та Threat (LOW/MEDIUM/HIGH).

У статичних сценаріях роботу контролера переводять у режим ASM_FIXED_PROFILE, коли FLC не використовують, а профіль задають фіксовано (наприклад, Max або Balanced) для порівняння з адаптивним режимом. Це створює умови для коректного зіставлення результатів експериментів «Adaptive Full Battery», «Adaptive Low Battery», «Fixed Max» та «Fixed Balanced» без зміни кодової бази.

Наприкінці кожної ітерації ASM-Agent записує оновлену політику в security_policy.json, а також дописує рядок у CSV-лог experiment_data.csv, після чого робить паузу на ASM_INTERVAL_SEC секунд до наступного циклу. Тривалість інтервалу опитування визначає дискретність адаптивного керування і може бути налаштована залежно від динаміки навантаження та вимог до швидкості реагування.

3.4.3 Алгоритм роботи сервісу прикладного програмного інтерфейсу та зв’язок із профілями

ASM API реалізує REST-інтерфейс, у якому кожен HTTP-запит до кінцевої точки (endpoint) ingest інтерпретують як передавання даних від IoT-пристрою до fog-шлюзу з подальшим криптографічним обробленням. Після прийому запиту сервіс визначає актуальний профіль захисту із застосуванням функції get_current_profile, що зчитує поле final_profile або base_profile із файлу security_policy.json, а в разі їхньої відсутності застосовує значення APP_DEFAULT_PROFILE зі змінної середовища. Послідовнісну діаграму оброблення запиту наведено на рис. 3.7.

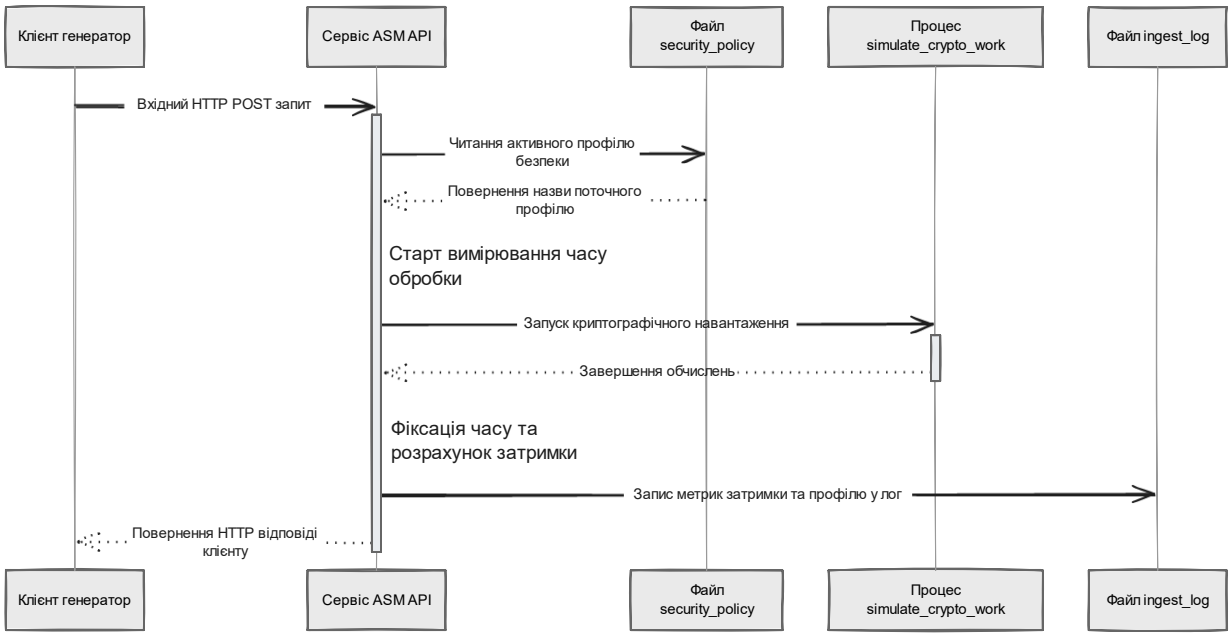

Рисунок 3.7 – Послідовнісна діаграма обробки HTTP-запиту в ASM API

Далі викликають процедуру simulate_crypto_work(profile), яка емулює криптографічне навантаження шляхом багаторазового виконання операцій гешування з різною інтенсивністю залежно від профілю. Для профілю max_security використовують великий ліміт ітерацій із застосуванням SHA-512, для профілю balanced – середню кількість ітерацій із SHA-256, для профілю survival – меншу кількість ітерацій із SHA-1, тоді як профіль safe_shutdown працює з найменшою кількістю ітерацій MD5, що моделює режим мінімального навантаження в умовах навмисно зниженого рівня безпеки. Такий вибір формує ієрархію криптографічних профілів від максимальної криптостійкості до ресурсоощадних, але слабких режимів.

Вибір таких профілів відображає загальні уявлення про компроміс між криптостійкістю та енергоспоживанням/затримкою для різних криптографічних примітивів, які наводять у роботах з оцінювання ефективності полегшених та класичних алгоритмів для IoT-пристроїв, де підкреслюють потребу знаходити баланс між силою захисту, часом виконання та ресурсними витратами [94].

Час оброблення запиту вимірюють із застосуванням високоточних таймерів (time.perf_counter), а результат конвертують у мілісекунди для подальшого аналізу затримки. Після завершення емульованого криптооброблення формують лог-запис у форматі JSON з полями ts (час сервера), device_id, profile, proc_time_ms, який

дописують до файлу ingest_log у вигляді рядків JSONL. Зібрані дані надалі агрегує ASM-Agent або зовнішні аналітичні інструменти для обчислення ковзної середньої затримки, що входить до формули нормованої затримки L_{norm} та інтегрального критерію якості J.

Відповідь клієнту формують у вигляді JSON-структури, що містить статус, профіль, який фактично застосовано, та виміряну затримку оброблення. Такий зворотний зв'язок дає змогу на етапі тестування зіставляти очікуваний рівень сервісу для різних профілів із реально спостережуваними затримками й, за потреби, коригувати ваги інтегрального критерію J або патерни FLC-правил.

3.4.4 Структури керуючих параметрів та конфігураційних матриць

Файл security_policy.json є ключовим механізмом обміну станом між ASM-Agent та ASM API й містить поля battery, base_profile, final_profile, threat_level, security_score, updated_at тощо. Поле battery зберігає поточний рівень заряду батареї у відсотках, поля base_profile/final_profile задають вихідний та фактично застосований профілі (з урахуванням рішень FLC або фіксованого режиму), threat_level відображає числову оцінку рівня загроз, а security_score містить обчислений FLC показник безпеки. Поле updated_at забезпечує відстеження актуальності політики та синхронізацію рішень між компонентами в умовах асинхронної роботи.

Файл threat_state.json має компактну структуру з полями threat_level, updated_at, max_device, device_stats, що формує можливість як для сценаріїв експерименту, так і для зовнішніх засобів виявлення аномалій задавати поточний рівень загрози та, за потреби, виділяти пристрій, який генерує найбільше навантаження. Цей механізм дає змогу реалізувати сценарії, де зростання RPS або виявлення DDoS-атаки інтерпретують як підвищення Threat Level, що надалі впливає на вибір криптопрофілю через FLC.

Параметри ваг цільового критерію та режимів роботи контролера задають через змінні середовища, які зчитують у конфігураційному модулі. Серед них

WEIGHT_LATENCY, WEIGHT_BATTERY, WEIGHT_SECURITY визначають значення ваг w_L, w_P, w_S, ASM_INITIAL_BATTERY встановлює початковий рівень заряду батареї для сценарію (наприклад, Full Battery чи Low Battery), ASM_INTERVAL_SEC задає інтервал опитування в секундах.

3.4.5 Структура телеметрії та вимоги до відтворюваності

CSV-лог experiment_data.csv, який веде ASM-Agent, містить структуровані записи з ключовими параметрами стану системи на кожній ітерації циклу адаптивного керування. Лог-файл ingest-запитів (ingest_log.json) містить окремі записи для кожного HTTP-запиту у вигляді JSON-рядків з полями ts (timestamp сервера), device_id, profile, proc_time_ms. На основі цього логу формують ковзні метрики середньої затримки та, за потреби, статистики за профілями, які використовують для порівняння сценаріїв Adaptive та Fixed у частині латентності. Стандартизований формат JSONL спрощує оброблення логів із застосуванням зовнішніх аналітичних інструментів, що відповідає практиці масштабованого моніторингу IoT- та fog-систем.

Docker-композитор (docker-compose.yml) фіксує склад сервісів, мережеві налаштування, спільні томи (volumes) для логів і конфігурацій, а також змінні середовища для кожного контейнера, що створює змогу запускати стенд однією командою на будь-якому ARM SoC. Основні параметри, серед яких ASM_INITIAL_BATTERY, ASM_INTERVAL_SEC, вагами цільового критерію та фіксованими профілями для базових сценаріїв, задають через змінні середовища (ENV) або конфігурації, що забезпечує повторюваність результатів та можливість детерміновано відтворювати сценарії з різними налаштуваннями без модифікації вихідного коду.

Формат логу наведено в табл. 3.6. Така структура логів забезпечує можливість подальшого аналізу компромісів між енергоспоживанням, продуктивністю й рівнем захисту за різних сценаріїв та наборів вагових коефіцієнтів.

Таблиця 3.6 – Формат CSV-логу ASM-Agent

Назва стовпця	**Одиниці вимірювання**	**Опис параметра**
timestamp	с (Unix time)	Поточна системна мітка часу на момент запису рядка.
battery	%	Поточний розрахунковий рівень заряду батареї вузла.
threat	Безрозмірна (0..1)	Інтегральний рівень загрози, отриманий від підсистеми моніторингу.
fuzzy_score	Безрозмірна (0..100)	Чітке числове значення виходу нечіткого регулятора FLC (Security Score) після дефазифікації.
profile	Рядок (Текст)	Лінгвістична назва поточного активного профілю безпеки (safe_shutdown, survival, balanced, max_security).
J	Безрозмірна	Значення інтегрального критерію якості, що відображає поточний компроміс між затримкою, зарядом та безпекою.
latency	мс	Середня затримка обробки запитів модулем ASM API за останній інтервал моніторингу.
profile_numeric	Дискретна (0..3)	Числовий ідентифікатор профілю (0 - safe, 1 - survival, 2 - balanced, 3 - max) для зручності побудови графіків.
real_cpu_load	%	Фактичне апаратне завантаження центрального процесора (CPU) fog вузла.
real_cpu_temp	°C	Апаратна температура центрального процесора (CPU thermal) вузла.

3.4.6 Інтеграція фреймворку в ширше середовище Інтернету Речей

Реалізований ASM-фреймворк можливо інтегрувати в реальне fog-середовище як надбудову над наявними шлюзовими сервісами. ASM API здатний виступати фронтенд-шлюзом для IoT-пристроїв, приймаючи дані через HTTP(S) або в ролі проміжного сервісу поруч з уже розгорнутими протоколами (наприклад, MQTT-брокером) з подальшим передаванням трафіку до хмарних сервісів. Схему інтеграції зображено на рис. 3.8.

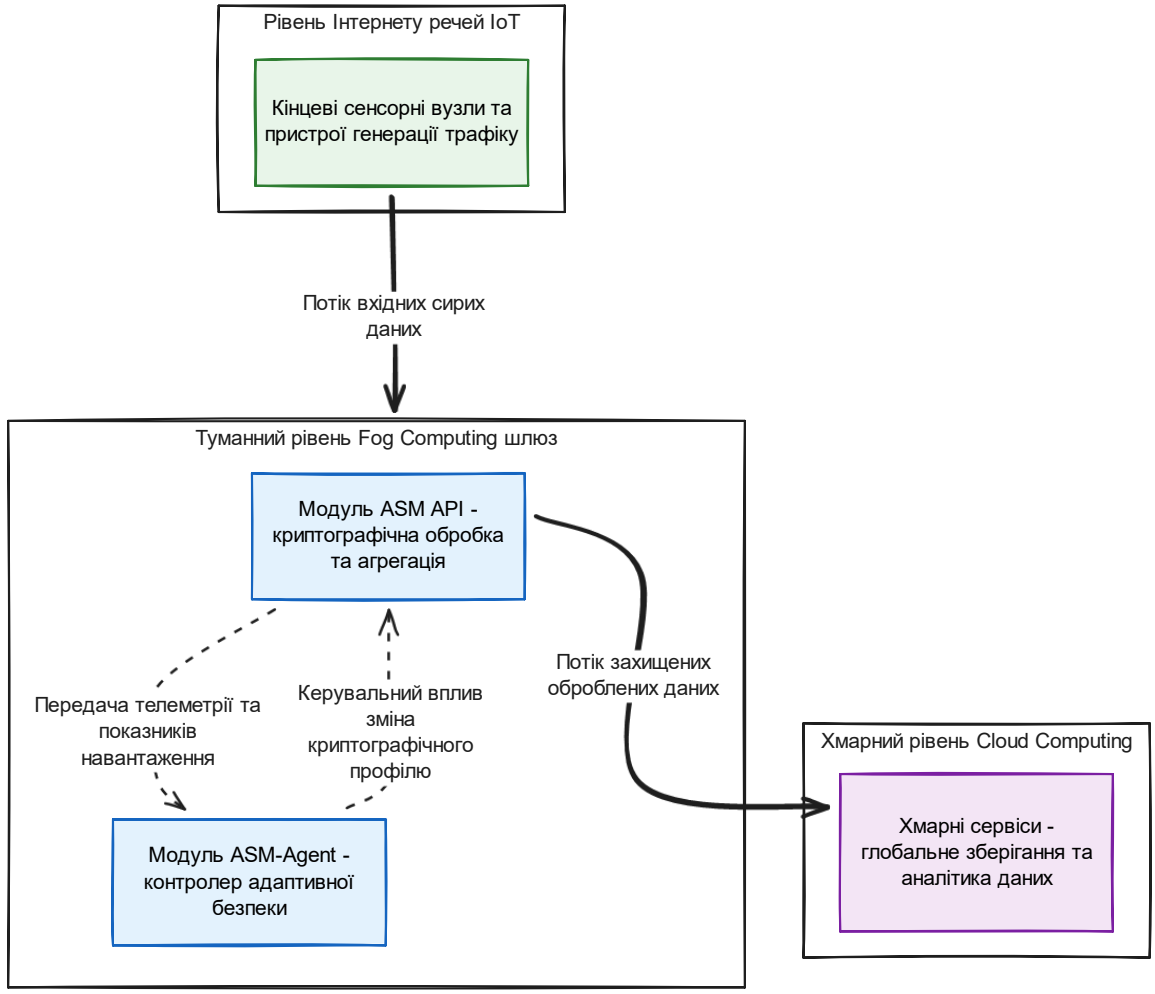

Рисунок 3.8 – Схема інтеграції ASM-фреймворку в fog/IoT-архітектуру

Водночас, ASM-Agent функціонує як окремий сервіс моніторингу та адаптивного керування криптопрофілем, який у промисловому (production) варіанті здатен взаємодіяти з реальними криптобібліотеками та TLS-/DTLS-стеками замість емульованого навантаження simulate_crypto_work.

У промисловому застосуванні замість демонстраційного модуля crypto_profiles.py, що використовує AES-GCM, ChaCha20-Poly1305, HMAC-SHA-256/512 та CRC32 у режимі емуляції, доцільно застосовувати апаратно прискорені криптобібліотеки або сертифіковані реалізації полегшених алгоритмів, які рекомендовано для обмежених пристроїв. Водночас, логіку FLC-контролера, структуру конфігураційних файлів та формат логів зберігають із мінімальними змінами, оскільки вони оперують абстрактними профілями й метриками, а не конкретними алгоритмами.

Висновки до розділу

У розділі розроблено метод енергоорієнтованого адаптивного керування криптографічним захистом fog-вузлів, в основу якого покладено архітектурну концепцію ASM-фреймворку. Доведено, що формування цілісного контуру керування на основі безперервного моніторингу ключових контекстних змінних – рівня заряду батареї та рівня кіберзагроз (Threat Level) – забезпечує плавне зниження рівня захисту криптографічного навантаження. Це дає змогу уникнути повної відмови шлюзу в умовах енергодефіциту зі збереженням мінімально прийнятного рівня безпеки та функціональної живучості вузла.

Центральним елементом запропонованого методу є розроблений нечіткий логічний контролер (FLC) типу Такаґі-Сугено (Takagi–Sugeno) нульового порядку. Обґрунтовано використання трапецієподібних та трикутних функцій належності в поєднанні з пороговою дискретизацією та механізмами стабілізації перемикань (запобігання брязкоту). Зазначений математичний апарат гарантує гладку й інтерпретовану поведінку системи під час обчислення агрегованої оцінки рівня безпеки (Security Score). Формалізована база правил та визначена множина криптографічних профілів (Safe, Survival, Balanced, Max) створюють можливість для контролера гнучко адаптувати захист відповідно до вимог щодо конфіденційності, цілісності та доступності (CIA), мінімізуючи розроблений інтегральний критерій якості J.

На основі аналізу наявних підходів встановлено, що запропонований метод вирішує актуальну проблему обмеженості засобів адаптивної безпеки для мобільного IoT. На відміну від відомих рішень, де нечітку логіку застосовують здебільшого для загального енергоменеджменту або оцінювання ризиків, у роботі реалізовано безпосереднє керування криптографічними режимами з урахуванням моделі батареї. Контейнеризована програмна реалізація фреймворку в Docker-оточенні з розподілом на компоненти ASM API та ASM-Agent забезпечує високу відтворюваність, керованість та переносимість розробленого рішення на платформи рівня ARM SoC.

Сформована математична й алгоритмічна база, а також створена програмна архітектура методу утворюють повноцінне підґрунтя для його впровадження на апаратних fog-платформах з обмеженими ресурсами. Водночас запропоновані теоретичні результати, політика профілювання та ефективність роботи нечіткого контролера потребують практичної верифікації. Відповідно, подальшим етапом дослідження є проведення натурних випробувань методу на спеціалізованому експериментальному HiL-стенді (Hardware-in-the-Loop) з метою порівняльного аналізу адаптивної стратегії з базовими статичними підходами.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ, РЕЗУЛЬТАТИ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ

4.1 Апаратно-програмний стенд і методика експериментів

У цьому розділі наведено результати експериментальної перевірки та порівняльного аналізу розробленого методу енергоорієнтованого адаптивного керування. Основні теоретичні положення, методика тестування на базі HiL-стенду та ключові експериментальні результати, що викладені в цьому розділі, були апробовані та опубліковані у праці [95].

4.1.1 Hardware-in-the-Loop

Підхід апаратного моделювання (Hardware-in-the-Loop (HiL)) для оцінювання енергоорієнтованих механізмів захисту у fog-вузлах базують на поєднанні реального апаратного середовища з програмними моделями підсистем, які складно або економічно недоцільно відтворювати фізично під час кожного циклу тестування (зокрема, моделі акумуляторної батареї, генерації мережевого трафіку та сценаріїв кіберзагроз). Така гібридна архітектура створює можливість достовірно досліджувати енергоспоживання, апаратне завантаження процесора, поведінку операційної системи та тепловий режим на реальному одноплатному комп’ютері, тоді як параметри зовнішнього середовища та кібератак змінюють програмно, що забезпечує високу контрольованість і відтворюваність експериментальних результатів.

Як базовий fog-вузол експериментального HiL-стенду використано одноплатний мікрокомп’ютер Raspberry Pi 5 із 64-бітним процесором ARM Cortex-A76 та об’ємом оперативної пам’яті (ОЗП) 8 ГБ. Ця апаратна платформа забезпечує достатній запас обчислювальних ресурсів для паралельної реалізації механізмів криптографічного захисту, нечіткого логічного контролера (FLC, Fuzzy Logic Controller) та сервісної логіки адаптивного менеджера безпеки (ASM, Adaptive Security Manager). Структурну схему розробленого апаратно-програмного стенду наведено на рис. 4.1.

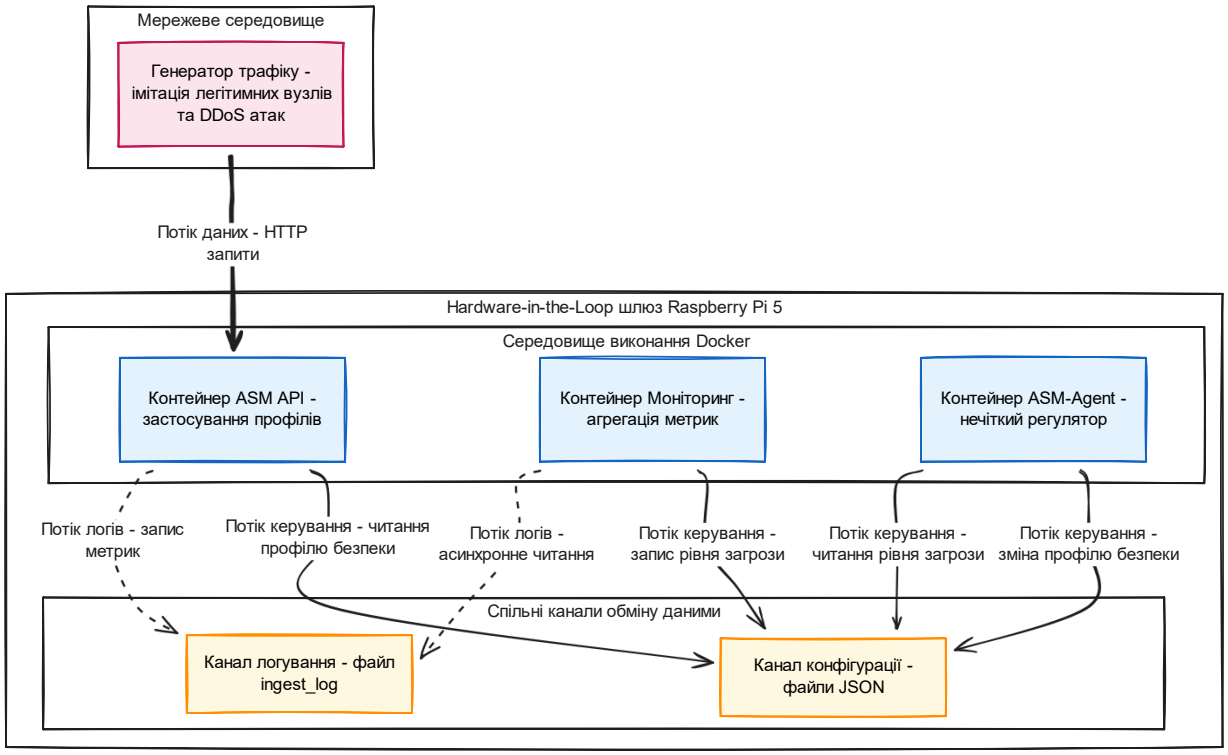

Рисунок 4.1 – Структурна схема HiL-стенду

Застосована архітектура на базі ядер Cortex-A76 формує можливість відтворити типові умови навантаження fog-рівня, за яких на одному фізичному вузлі одночасно функціонують мережевий стек, підсистема захисту та служба керування політиками безпеки. Оскільки зазначений процесор має апаратну підтримку криптографічних інструкцій, для забезпечення можливості фіксації значущих енергетичних витрат на мережевому рівні в межах короткочасних експериментів, вузол виконує синтетичні імітаційні криптографічні операції (контрольоване ітеративне гешування) відповідно до обраного профілю захисту, зберігаючи водночас суттєву чутливість до енергетичних та обчислювальних обмежень.

HiL-стенд реалізує енергетичну модель fog-вузла у вигляді програмної «віртуальної батареї», стан якої (заряд батареї) безперервно оновлюють на основі фактичного апаратного завантаження центрального процесора (CPU load), заданого інтервалу дискретизації та емпіричних параметрів моделі розряду. Модель динаміки заряду батареї інтегровано в компонент ASM-Agent і реалізовано як дискретне наближення: нове значення заряду обчислюють як функцію попереднього стану та

ефективного струму розряду («дренажу»), який лінійно залежить від інтенсивності роботи процесора і визначається калібрувальними константами r_{min} та r_{max}.

Концептуально архітектуру HiL-стенду орієнтовано на об'єктивне оцінювання запропонованого адаптивного механізму, у якому вибір криптографічного профілю здійснює нечіткий регулятор на основі моніторингу трьох ключових параметрів: Battery Level (рівень заряду акумулятора), Threat Level (поточний рівень кіберзагрози) та інтегральної оцінки безпеки (Security Score). Результати апаратних вимірювань завантаження CPU, затримок оброблення запитів та динаміки деградації заряду батареї під час експериментів формують емпіричну базу для розрахунку цільової функції (штрафу) та комплексного порівняння ефективності адаптивного керування зі статичними стратегіями захисту.

4.1.2 Апаратна архітектура стенду

Апаратну основу експериментального HiL-стенду (апаратна схема якого наведена на рис. 4.2) формує високоінтегрована електронна платформа на базі мікрокомп’ютера Raspberry Pi 5.

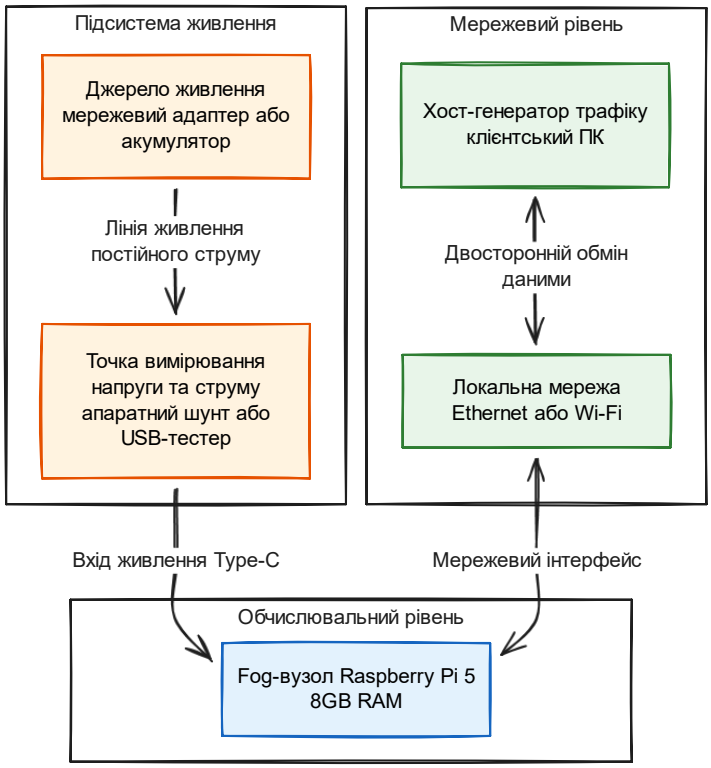

Рисунок 4.2 – Апаратна схема HiL-стенду

З позиції електронної інженерії, цей вузол є репрезентативною fog-системою, архітектура якої об’єднує високопродуктивну систему на кристалі Broadcom BCM2712 та спеціалізовану підсистему керування живленням Renesas Dialog DA9091. Використання дискретної мікросхеми керування живленням забезпечує багатоканальне перетворення напруг (DC-DC) для живлення ядер процесора та периферії. Така апаратна топологія створює можливість у перспективі здійснювати прецизійне калібрування моделі розряду шляхом фізичного вимірювання струмів і напруг на шунтах ліній живлення в умовах переходу на автономне акумуляторне джерело. Детальні технічні характеристики платформи наведено в табл. 4.1.

Таблиця 4.1 – Технічні характеристики Raspberry Pi 5

Апаратна підсистема / Компонент	Специфікація
Система на кристалі (SoC)	Broadcom BCM2712
Процесорне ядро (CPU)	4-ядерний ARM Cortex-A76 (64-bit) @ 2.4 ГГц
Контролер вводу/виводу (Південний міст)	Raspberry Pi RP1
Оперативна пам'ять	8GB LPDDR4X-4267
Підсистема управління живленням	Renesas Dialog DA9091
Вимоги до живлення	5V / 5A DC (через USB-C PD)
Інтерфейс загального призначення (GPIO)	40-pin роз'єм (сумісний з попередніми версіями)
Високошвидкісні шини	1 × PCIe 2.0 x1 (через FPC роз'єм)
Мережеві інтерфейси	Gigabit Ethernet, Wi-Fi 802.11ac, Bluetooth 5.0 / BLE
Пам'ять програм / Даних	Слот MicroSD

Фото мікрокомп’ютера Raspberry Pi 5 представлено на рис 4.3.

Рисунок 4.3 – Одноплатний мікрокомп'ютер Raspberry Pi 5

Загальний вигляд апаратно-програмного стенду для проведення експериментальних досліджень та верифікації методу адаптивного керування захистом представлено на рис. 4.4.

Рисунок 4.4 – Загальний вигляд дослідного стенду

Комутацію fog-вузла із зовнішнім хостом (ПК або ноутбуком) для конфігурування сценаріїв трафіку та віддаленого керування Docker-контейнерами здійснюють через локальну мережу. Для забезпечення детермінованості вимірювань затримок обрано використання дротового фізичного рівня (PHY) інтерфейсу Gigabit Ethernet. Це інженерне рішення мінімізує вплив флуктуацій і завад радіочастотного тракту безпроводового каналу Wi-Fi та дає змогу сфокусувати вимірювання виключно на апаратних затримках, зумовлених обчислювальним навантаженням криптографічних примітивів та контролера FLC. Маршрутизацію тестового навантаження забезпечує високопродуктивний мережевий шлюз, апаратні параметри якого зведено в табл. 4.2.

Таблиця 4.2 – Технічні характеристики маршрутизатора ASUS TUF Gaming BE3600

Апаратна підсистема / Компонент	Специфікація
Система на кристалі (SoC) / Обчислювальне ядро	4-ядерний процесор Broadcom @ 2.0 ГГц (обчислювальна потужність ~20K DMIPS)
Радіочастотний тракт (RF) та модуляція	Wi-Fi 7 (802.11be), апаратна підтримка 4096-QAM (Quadrature Amplitude Modulation)
Архітектура безпроводового зв'язку (MAC/PHY)	Multi-Link Operation (MLO) у діапазонах 2.4 ГГц (до 688 Мбіт/с) та 5 ГГц (до 2882 Мбіт/с), 2x2 MIMO
Підсистема пам’яті	1 GB DDR4 RAM256 MB Flash-пам'яті (NAND)
Топологія проводового вводу/виводу	1 × 2.5 GbE (WAN/LAN) PHY-контролер4 × 1 GbE (LAN) BaseT
Інтерфейс розширення та резервування зв'язку	1 × USB 3.2 Gen 1 (з підтримкою USB 3G/4G/5G стільникових модемів)
Енергетична підсистема та тепловідвід	Живлення: 12V / 2A DC (макс. споживання 24 Вт). Алюмінієві радіатори з нанокарбоновим покриттям.

Функціонування електронної компонентної бази fog-вузла в реальних умовах спричиняє динамічну зміну споживаної потужності. Відповідно до законів фізики напівпровідників, динамічна потужність розсіювання в КМОН-структурах (Complementary Metal-Oxide-Semiconductor (CMOS)) мікропроцесора є прямо пропорційною тактовій частоті та квадрату напруги живлення ($P \propto f \cdot V^2$). У межах стенду процесор ARM Cortex-A76 працює під керуванням механізмів динамічного масштабування напруги та частоти, які вбудовані в операційну систему. Вони

безпосередньо впливають на тепловий режим (термальний тротлінг) та енергоспоживання ядра. Саме ці фізичні апаратні процеси є базисом для програмної моделі ASM-Agent, де ефективний струм розряду апроксимують через калібрувальні коефіцієнти r_{min} та r_{max}. Ці константи відображають відповідно статичне (струми витоку) та динамічне (комутаційне) енергоспоживання напівпровідникового кристала під навантаженням, перетворюючи логічний алгоритм захисту на фізично вимірний енергетичний процес.

Для забезпечення метрологічної точності програмної моделі віртуальної батареї в ASM-Agent базові параметри розряду r_{min} та r_{max} потребують апаратного калібрування. З позицій електронної інженерії таке калібрування є включенням прецизійного шунтового резистора з малим температурним коефіцієнтом опору в розрив лінії живлення постійного струму (5V DC). Вимірювання падіння напруги на шунті із застосуванням зовнішнього аналого-цифрового перетворювача (АЦП) високої роздільної здатності (наприклад, інтегральних датчиків струму та потужності архітектури INA219/INA226) створює можливість отримати реальні осцилограми споживаного струму. Інтегрування цих миттєвих значень струму в часі дає змогу точно визначити енергетичну вартість кожної криптографічної імітації (Max, Balanced, Survival) та перенести ці фізичні величини у програмну модель у вигляді вагових коефіцієнтів деградації заряду.

Крім того, критичним фактором для енергообмежених fog-вузлів є керування тепловим розсіюванням напівпровідникового кристала (Thermal Management). В умовах тривалого застосування профілю Max (імітація AES-256-GCM) завантаження ARM-ядра наближається до 100%, що спричиняє експоненційне зростання температури SoC Broadcom BCM2712. Відповідно до фізики напівпровідників, підвищення температури T зумовлює експоненційне зростання паразитних струмів витоку ($I_{leak} \propto e^{T}$) у КМОН-транзисторах, що додатково збільшує статичне енергоспоживання плати незалежно від корисного навантаження. У разі досягнення критичної температури (близько 80–85 °C) підсистема живлення ініціює термальний тротлінг – примусове зниження тактової частоти й напруги ядра (DVFS). Зі зниженням частоти фізичний час виконання контрольних ітерацій гешування зростає,

що в результаті фіксують на рівні ASM-фреймворку як різке збільшення затримки оброблення пакетів (latency). Отже, запропонований адаптивний алгоритм FLC, уникаючи перманентного використання профілю Max, виконує не лише функцію енергозбереження, а й запобігає деградації напівпровідникових структур унаслідок теплового перевантаження.

4.1.3 Мікропрограмна архітектура та ізоляція процесів тестового середовища

Програмну архітектуру HiL-стенду побудовано за мікросервісною контейнерною моделлю з використанням середовища віртуалізації на рівні операційної системи (Docker). Це забезпечує жорстку ізоляцію компонентів, контроль версій вбудованого програмного забезпечення та високу відтворюваність експериментальних результатів у гетерогенних середовищах. На апаратній платформі fog-вузла розгортають щонайменше два ізольовані контейнери: модуль оброблення трафіку (ASM API) та модуль керування безпекою і енергоспоживанням (ASM-Agent). Взаємодію між ними здійснюють через спільні конфігураційні файли та журнали подій, змонтовані як загальні томи пам'яті (Docker Volumes), що імітує архітектуру апаратно-незалежної шини даних.

Контейнер ASM API реалізує мережевий HTTP-інтерфейс на базі фреймворку FastAPI та приймає вхідні запити від зовнішнього генератора трафіку, відтворюючи поведінку fog-шлюзу, до якого підключають множину віддалених IoT-пристроїв. Для кожного вхідного запиту API викликає функцію simulate_crypto_work(). Оскільки мікропроцесор ARM Cortex-A76 оснащений апаратними криптографічними прискорювачами, пряме застосування еталонних алгоритмів (AES-256-GCM, ChaCha20-Poly1305 тощо, що описані в модулі crypto_profiles.py) унеможливлює фіксацію мікросекундних енергетичних витрат на мережевому рівні. Тому функція simulate_crypto_work() реалізує метод імітаційного синтетичного навантаження: для кожного запиту генерують обчислювальний цикл багаторазового гешування (наприклад, 25 000 ітерацій SHA-512 для профілю Max). Це дає змогу фізично

завантажити арифметико-логічний пристрій (АЛП) процесора пропорційно до обчислювальної складності цільового криптографічного профілю, визначеного у файлі security_policy.json.

Контейнер ASM-Agent функціонує як системний супервізор. Він інкапсулює контролер нечіткої логіки (FLC), програмну модель розряду акумуляторної батареї та модуль ретроспективного розрахунку інтегрального критерію якості J. Алгоритм фазифікації оперує вхідними фізичними та логічними змінними: Battery Level (поточний заряд батареї), Threat Level (рівень кіберзагрози з файлу threat_state.json) та Security Score (дискретні еквіваленти 10, 30, 60, 90 для відповідних профілів). FLC використовує трапецієподібні та трикутні функції належності для лінгвістичних термів (LOW, MEDIUM, HIGH) і реалізує базу експертних правил для вибору оптимального енергобезпекового профілю.

Організацію міжпроцесної взаємодії (Inter-Process Communication (IPC)) реалізовано через спільні конфігураційні артефакти. Файл security_policy.json виступає як буфер стану: ASM-Agent із заданою частотою дискретизації оновлює фізичні та логічні параметри системи (battery, threat_level, base_profile, final_profile), тоді як ASM API асинхронно зчитує актуальний final_profile для конфігурування обчислювального навантаження АЛП. Файл threat_state.json містить поточний індикатор загрози та часову мітку оновлення (timestamp). Така асинхронна архітектура обміну даними через спільну пам'ять мінімізує системні накладні витрати на синхронізацію та дає змогу легко модернізувати механізм керування (наприклад, інтегрувати апаратні датчики струму I2C/SPI замість програмної моделі розряду) без порушення монолітності системи.

Генератор мережевого трафіку реалізовано як зовнішній апаратно-програмний вузол, що генерує HTTP-запити до ASM API із заданим профілем навантаження (стаціонарна інтенсивність, імпульсні сплески, DDoS-подібні шторми запитів). Крім того, зовнішній оркестратор має можливість синхронно модифікувати параметр Threat Level у файлі threat_state.json, імітуючи функціонування зовнішньої системи виявлення вторгнень (IDS) та відтворюючи ескалацію кіберзагроз у реальному часі для перевірки динаміки відпрацювання алгоритму керування.

4.1.4 Сценарії експериментів

Експериментальну валідацію запропонованого методу здійснено в межах чотирьох базових сценаріїв, кожен з яких мав фіксовану тривалість 120 с. Зведені характеристики та початкові апаратні умови цих сценаріїв наведено в табл. 4.3.

Таблиця 4.3 – Характеристика сценаріїв експериментального навантаження fog-вузла

Назва сценарію	Режим керування	Початковий рівень заряду	Політика вибору криптографічного профілю
Adaptive Full Battery	Адаптивний	100%	Динамічно (визначається контролером FLC)
Adaptive Low Battery	Адаптивний	35%	Динамічно (визначається контролером FLC)
Fixed Max	Неадаптивний (статичний)	Будь-який	Жорстко фіксований на рівні Max
Fixed Balanced	Неадаптивний (статичний)	Будь-який	Жорстко фіксований на рівні Balanced

У сценаріях з адаптивним керуванням (Adaptive Full Battery та Adaptive Low Battery) активний профіль оброблення визначав нечіткий логічний контролер (FLC) виключно на основі поточних значень заряду батареї та логічного рівня загрози (Threat Level). Натомість у сценаріях Fixed Max та Fixed Balanced застосовували неадаптивні стратегії, де обраний рівень імітаційного криптографічного навантаження залишався незмінним незалежно від фізичного стану батареї чи інтенсивності мережевих атак.

Для забезпечення об'єктивності порівняльного аналізу в усіх сценаріях застосовано ідентичний часовий профіль мережевого навантаження, який складався з трьох фаз:

1) 0–40 с: Фаза фонового режиму роботи з номінальною інтенсивністю запитів, що відповідала низькому рівню кіберзагрози (Threat Level = LOW).

2) 40–80 с: Фаза пікового навантаження, яка імітувала масовану DDoS-атаку та відповідала високому рівню загрози (Threat Level = HIGH). У цей період інтенсивність вхідного трафіку (RPS) експоненційно зростала.
3) 80–120 с: Фаза релаксації та відновлення до фонового режиму роботи після завершення атаки (Threat Level = LOW).

Переходи між інтервалами генерували детерміновано за часом, а параметри зовнішнього генератора навантаження залишалися абсолютно ідентичними для всіх запусків. Це гарантувало метрологічну відтворюваність експериментів та коректність подальшого інтегрування енергетичних витрат.

Для мінімізації впливу перехідних апаратних процесів перед кожним запуском систему приводили в суворо регламентований вихідний стан: ініціалізували системні лічильники, встановлювали початковий заряд батареї B_0 відповідно до обраного сценарію, очищували кеші оперативної пам'яті, а мікропроцесор проходив етап температурної стабілізації (охолодження кристала до номінальних значень для запобігання передчасному термальному тротлінгу). Кожен сценарій повторювали 5 разів, після чого результати вимірювань агрегували та піддавали статистичному усередненню.

Збір експериментальних даних забезпечував ізольований програмний компонент моніторингу (ASM-Agent). Він синхронно реєстрував фізичні та логічні системні метрики: апаратне завантаження обчислювального ядра (CPU load), інтенсивність оброблення запитів, середню затримку на мережевому рівні (latency), динаміку деградації заряду батареї, а також фіксував активний криптографічний профіль. Крім того, на кожному такті здійснювали розрахунок інтегрального критерію якості J для цілей подальшого ретроспективного аналізу. Отримані масиви даних зберігали в журналах формату CSV із фіксованим кроком дискретизації вимірювань (близько 1 с), що є достатнім для точного відтворення динаміки досліджуваних теплових та енергетичних процесів.

4.1.5 Метрики ефективності, збір даних та синхронізація логів

Комплексне оцінювання апаратно-програмної ефективності запропонованого адаптивного механізму захисту здійснюють на основі багатовимірного набору метрик, що об’єктивно відображають енергетичний стан, завантаження обчислювальних ресурсів, тепловий режим, мережеві затримки обслуговування та рівень криптографічної стійкості (табл. 4.4).

Таблиця 4.4 – Огляд зібраних метрик

Метрика (Параметр)	Одиниці вимірювання	Основне джерело даних (Файл)	Період дискретизації / Характер запису
Battery Level	%	experiment_data.csv, security_policy.json	Дискретно, кожні 0.5–10с (визначається ASM_INTERVAL_SEC)
EnergyDelta (Витрачена енергія)	%	Розрахункова (на основі experiment_data.csv)	Агрегований показник (різниця за повний сеанс тривалістю 120 с)
AvgCPU (Апаратне завантаження CPU)	%	experiment_data.csv	Дискретно, кожні 0.5–10с (базовий інтервал супервізора)
AvgLatency (Середня затримка оброблення)	мс	Первинно: ingest.logАгреговано: experiment_data.csv	Сирі дані: асинхронно (для кожного запиту) Усереднено: кожні 0.5–10 с
AvgSecurityScore (Чіткий рівень захисту)	Безрозмірна (0..100)	experiment_data.csv, security_policy.json	Дискретно, кожні 0.5–10с (базовий інтервал супервізора)
Threat Level (Рівень загрози)	Безрозмірна (0..1)	threat_state.json, experiment_data.csv	Оновлення: кожні 10с Зчитування: кожні 0.5–10 с
Температура напівпровідникового кристала (CPU Temp)	°C	experiment_data.csv	Дискретно, кожні 0.5–10 с (базовий інтервал супервізора)

Енергетичні метрики охоплюють поточний заряд батареї, а також інтегральний показник витраченої енергії (ΔE) за цикл експерименту, що інтерпретують як фізичну різницю між початковим та кінцевим значенням заряду батареї для конкретного сценарію експлуатації.

Метрики апаратного завантаження базують на безперервному системному моніторингу із застосуванням кросплатформних засобів ядра операційної системи (зокрема, бібліотеки psutil). Це створює можливість прецизійно фіксувати відсоток утилізації обчислювальних конвеєрів мікропроцесора та температурний режим напівпровідникового кристала (CPU temperature) на fog-вузлі. Отримані телеметричні дані використовують як базовий вхідний параметр для енергетичної моделі деградації заряду батареї, а також для кількісного аналізу відносної енергетичної ефективності кожного імітаційного криптопрофілю (Safe, Survival, Balanced, Max) з огляду на тепловиділення тепловиділення та витрати заряду акумулятора.

Затримку оброблення запитів (latency) вимірюють як фактичний час виконання функції синтетичного навантаження simulate_crypto_work() для кожного вхідного HTTP-запиту в компоненті ASM API і фіксують у мілісекундах у первинному журналі ingest.log. Модуль ASM-Agent агрегує ці дискретні вимірювання із застосуванням ковзного часового вікна (розмір якого визначають параметром Δt) та обчислює середнє значення затримки (L_{avg}). Цей показник надалі застосовують для математичного розрахунку нормалізованої складової L_{norm} у цільовому критерії.

Оцінку рівня безпеки формалізують через чіткий показник Security Score, який набуває фіксованих дискретних значень (10, 30, 60 або 90) залежно від обраного контролером профілю (Safe, Survival, Balanced або Max відповідно). Цей показник застосовують у нормалізованому вигляді (S_{norm}) у формулі інтегрального критерію якості J, математичну модель якого детально обґрунтовано в розділі 2.

Методикою збору експериментальних даних є паралельне журналювання на двох архітектурних рівнях. Модуль ASM API асинхронно записує параметри кожного мережевого транзакційного запиту у файл ingest.log у форматі JSONL (часова мітка, активний профіль, час апаратного оброблення, розмір корисного навантаження). Водночас супервізорний модуль ASM-Agent формує зведений CSV-журнал

experiment_data.csv з періодичними агрегованими записами стану системи: поточний заряд батареї, Threat Level, Security Score, активний профіль, розрахункове значення інтегрального критерію J, середня затримка оброблення, рівень завантаження та температура процесора.

Синхронізацію різнорідних масивів даних здійснюють на основі єдиної часової шкали (шляхом фіксації часових міток у форматі UNIX-часу). Такий підхід забезпечує коректне просторово-часове зіставлення телеметрії окремих мережевих запитів із глобальними показниками стану електронної системи, формуючи змогу будувати узгоджені часові діаграми розряду, тепловиділення, оцінки J та роботи алгоритму керування в подальших підрозділах.

4.1.6 Оцінка похибок та обмежень експериментальної методики

Експериментальна методика апаратно-програмного моделювання на базі HiL-стенду об'єктивно містить низку джерел метрологічних похибок, які безпосередньо впливають на точність оцінки енергоспоживання та часових метрик продуктивності. Одним із фундаментальних факторів є дискретність оновлення показника заряду батареї з фіксованим періодом Δt. Відповідно до теорії цифрового оброблення сигналів, це зумовлює виникнення похибки квантування (quantization error) змін заряду батареї в часі. Такий процес діє як фільтр низьких частот і потенційно згладжує короткочасні (високочастотні) піки енергетичного навантаження, зумовлені імпульсним мережевим трафіком.

Другим суттєвим джерелом апаратної похибки є спрощений характер застосованої програмної енергетичної моделі. Використана апроксимація встановлює лінійну залежність ефективного струму розряду батареї від ступеня завантаження обчислювального ядра (CPU load) у межах відкаліброваних констант r_{min} та r_{max}. Проте така математична модель не враховує нелінійних розрядних характеристик реальних електрохімічних джерел живлення, а також динамічного енергоспоживання периферійних контролерів, кеш-пам'яті різних рівнів, радіочастотних мережевих інтерфейсів та втрат на перетворення напруги в мікросхемі керування живленням.

Використання програмної батареї замість прямого інструментального вимірювання струму та напруги із застосуванням прецизійних вимірювальних шунтів та аналого-цифрових перетворювачів (АЦП) обмежує точність оцінювання абсолютних значень енергоспоживання у Ват-годинах (Вт·год). Однак, оскільки в межах дослідження систематична похибка моделі є константною для всіх тестових запусків, це створює можливість проводити високоточні відносні (компаративні) порівняння енергоефективності між адаптивним та статичними профілями керування за ідентичних умов.

Додатковий внесок у сумарну похибку вимірювань робить часовий джитер (jitter) операційної системи загального призначення. Затримка оброблення мережевих запитів (latency) залежить не лише від часу виконання синтетичних криптографічних ітерацій в арифметико-логічному пристрої (АЛП), а й від накладних витрат планувальника процесів (context switching), ієрархічного кешування та поточного стану буферів мережевого стека.

Для мінімізації впливу зазначених стохастичних похибок у HiL-методиці застосовано комплексний інженерний підхід. Він охоплює апаратну та програмну стабілізацію середовища виконання: ізоляцію процесів у Docker-контейнерах, мінімізацію фонових системних служб, використання детермінованого дротового Gigabit Ethernet з'єднання замість Wi-Fi, а також температурну стабілізацію напівпровідникового кристала мікропроцесора перед кожним запуском. Паралельно застосовано емпіричне калібрування базових параметрів r_{min} і r_{max} під контрольованим навантаженням. Крім того, використання алгоритмів ковзного усереднення (moving average) для показників latency та CPU-навантаження на достатньо широких часових вікнах дає змогу ефективно нівелювати високочастотний флуктуаційний шум вимірювань, забезпечуючи достовірність отриманих результатів.

4.1.7 Обґрунтування обсягу експериментальної вибірки

Експериментальну методологію вимірювання споживання енергії вузлом туманних обчислень, що функціонує на базі мікрокомп'ютера Raspberry Pi 5,

сформовано на основі сучасних підходів до профілювання електронної апаратури. Для збору даних замість зовнішніх апаратних вимірювачів застосовують внутрішню телеметрію операційної системи. Цей підхід забезпечує зчитування метрик з інтегрального контролера живлення із фіксованою частотою дискретизації. Достовірність використання внутрішніх вимірювачів живлення підтверджують дослідження щодо калібрування on-chip сенсорів на Edge-пристроях, де показано, що після калібрування внутрішніх сенсорів можна досягти високої точності оцінок енергоспоживання [96]. Також подібну практику збору системних часових рядів (завантаження мікропроцесора, використання пам'яті, тепловиділення) успішно застосовують для аналізу продуктивності апаратних вузлів Інтернету речей (IoT, Internet of Things) [97].

Для усунення впливу випадкового апаратного шуму здійснюють статистичну обробку результатів. Кожен експериментальний сценарій має жорстко фіксовану тривалість за часом, що становить 120 секунд. Застосування фіксованих часових вікон є поширеною практикою для побудови енергетичних моделей обчислювальних пристроїв [98]. Згідно з розробленою методикою, вимірювання повторюють n=5 разів для кожної конфігурації, після чого агрегують результати через обчислення середнього значення та стандартного відхилення.

Обґрунтування застосування обмеженої вибірки (n=5) ґрунтують на високій апаратній стабільності платформи Raspberry Pi та детермінованому характері робочого навантаження. За умов ідентичного виконання тестових сценаріїв дисперсія енергоспоживання між окремими запусками прямує до нуля. Як з'ясовано у сучасних дослідженнях з енергопрофілювання IoT-мереж, варіативність між запусками є істотно нижчою за ефект від зміни режиму роботи [99], у межах дослідження це дає змогу обмежитися невеликою кількістю повторень. Стабільність форми профілю потужності продемонстровано також в апаратних дослідженнях Edge-систем з профілюванням потужності, де повторювані вимірювання дають відтворювану базову поведінку мікропроцесора [100]. Крім того, аналіз варіабельності енергоспоживання доводить, що головний внесок у точність репрезентації робить збільшення тривалості одного вимірювального вікна (120 с), а не множення кількості незалежних повторень

[101]. З огляду на це, п’яти ітерацій вистачає для швидкої збіжності середнього значення, а подальше збільшення числа прогонів не зумовить підвищення наукової значущості експерименту.

4.2 Результати експериментів

4.2.1 Енергоспоживання та автономність fog-вузла

Криві залежності рівня заряду батареї від часу для кожного експериментального сценарію відтворювали з використанням програмної моделі батареї, математичний опис якої наведено у формулі (3.1). Часові діаграми активних профілів формували шляхом проєктування на вісь часу послідовності станів безпеки (Safe, Survival, Balanced, Max) із журналів компонента ASM-Agent, які було синхронізовано з динамікою рівня заряду (Battery Level) та рівня загрози (Threat Level). Застосований спосіб візуалізації створює можливість дослідити алгоритм роботи нечіткого логічного контролера (FLC) під час інтеграції даних про енергетичний стан вузла та рівень загроз, а також оцінити стабільність функціонування системи в кожному з режимів.

У сценарії «Adaptive Full Battery» з початковим рівнем заряду батареї близько 100% вхідна змінна Battery Level протягом усього експерименту залишається в області HIGH (заряд не опускається нижче 80%). Тому поведінку системи та вибір профілів визначає зміна рівня кіберзагрози Threat Level. На початковому етапі (0–40 с) в умовах фонового трафіку контролер нечіткої логіки (FLC) призначає системі профіль Balanced. За таких умов вихідний показник рівня безпеки (Security Score) формують у діапазоні, що генерує підвищене, але не максимальне інтегральне навантаження на центральний процесор.

Як показано на рис. 4.5, у фазі імітації масованої атаки (40–80 с), коли Threat Level стрімко зростає до рівня HIGH, FLC обґрунтовано ініціює перехід системи до найвищого рівня захисту – профілю Max. Відповідно до бази правил, наявність достатнього енергоресурсу дає змогу вузлу застосовувати ресурсомісткі криптографічні імітації для протидії загрозі. Після завершення атаки (починаючи з 80

с) та нормалізації загрозової обстановки до рівня LOW, FLC автоматично повертає систему до режиму Balanced, що спричиняє зменшення інтенсивності розряджання. За результатами вимірювань, інтегральна частка витраченої енергії (EnergyDelta) для цього сценарію становить 15,53%.

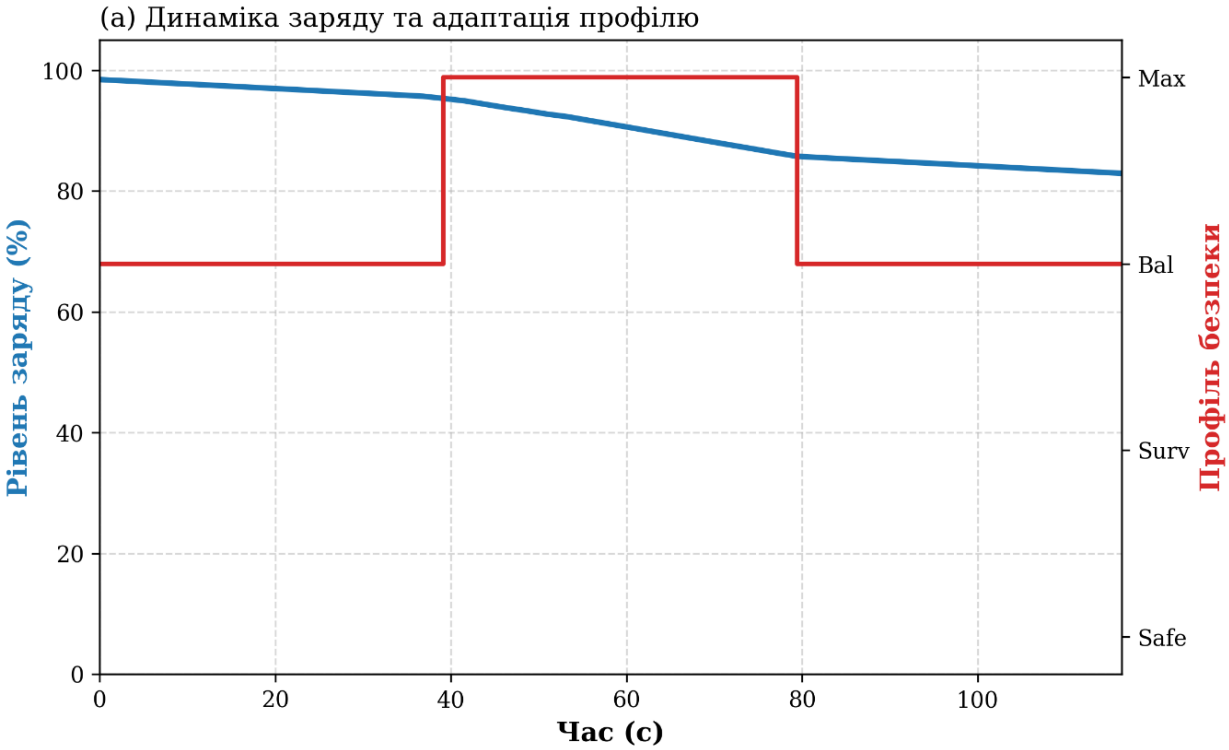

Рисунок 4.5 – Графік енергоспоживання сценарію Adaptive Full Battery

У сценарії «Adaptive Low Battery» початковий заряд батареї встановлено на критичному рівні близько 35%, що репрезентує умови експлуатації енергообмеженого fog-вузла в загрозовому середовищі. Вхідну змінну Battery Level контролер FLC ідентифікує як LOW. Відповідно до бази правил, це спричиняє швидкий перехід системи до профілю Survival (імітація ChaCha20-Poly1305) з ігноруванням високого показника Threat Level у фазі атаки. Як показано на рис. 4.6, в умовах наближення до критичного рівня заряду батареї контролер безальтернативно утримує полегшений профіль, успішно реалізуючи концепцію керованої деградації.

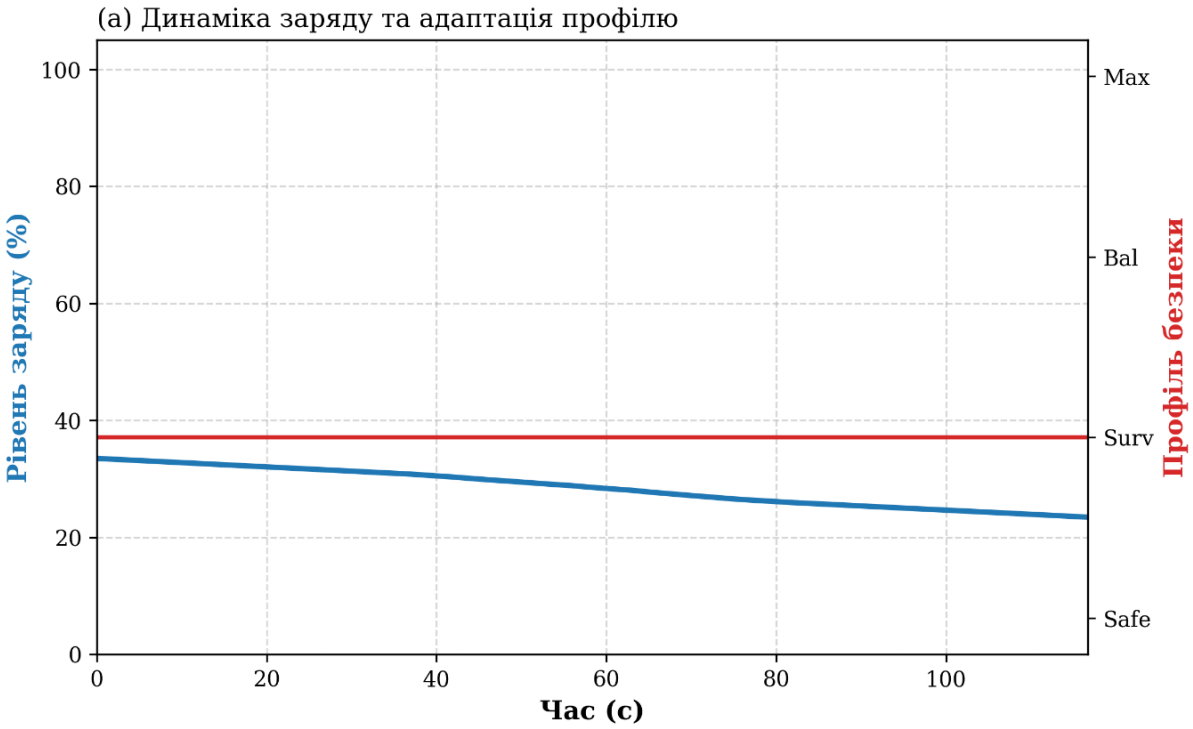

Рисунок 4.6 – Графік енергоспоживання сценарію Adaptive Low Battery

Застосування такого профілю дало змогу суттєво зменшити інтенсивність розряджання: показник EnergyDelta становить 10,04% за цикл експерименту (табл 4.5) за умови зниження середнього навантаження CPU. Порівняно зі статичним сценарієм Fixed Max, витрати енергії зменшилися на 40%, що математично екстраполюють як збільшення часу автономної роботи вузла під навантаженням в 1,7 раза. Рівень розряду для сценаріїв Adaptive Full Battery та Adaptive Low Battery за трьох наборах ваг представлено на табл. 4.5.

Таблиця 4.5 – Рівень розряду для двох сценаріїв за трьох наборах ваг

Набір вагових коефіцієнтів	Сценарій	Рівень розряду (%)
«Balanced»	Adaptive Full Battery	15.527
	Adaptive Low Battery	10.039
«Security»	Adaptive Full Battery	15.381
	Adaptive Low Battery	9.945
«Energy»	Adaptive Full Battery	14.816
	Adaptive Low Battery	9.514

Застосування порогового відображення Security Score на дискретні профілі виконує функцію механізму запобігання брязкоту (anti-chattering). Це забезпечує наявність відносно довгих стаціонарних ділянок роботи, що знижує накладні витрати на переналаштування криптографічних механізмів. У результаті FLC уникає осциляцій між профілями в умовах незначних флуктуацій Threat Level або заряду батареї, підтримуючи стабільні стани системи.

4.2.2 Апаратне завантаження процесора

Середню затримку оброблення запитів (latency) обчислюють на основі журналів компонента ASM API, де для кожного вхідного HTTP-запиту фіксують час виконання proc_time_ms. Застосування ковзного середнього дає змогу виділити довгострокові тренди та мінімізувати вплив короткочасних флуктуацій.

Рівень апаратного завантаження процесора (CPU load) визначають через системний моніторинг обчислювального ядра Raspberry Pi 5 у відсотковому вираженні, що узгоджується з метриками продуктивності для енергообмежених IoT-вузлів. Динаміка цих показників безпосередньо характеризує обчислювальну складність поточного криптографічного профілю та інтенсивність вхідного трафіку. Середні значення завантаження CPU для експериментів із режимами Adaptive Full Battery та Adaptive Low Battery наведено в табл. 4.6.

Таблиця 4.6 – Середні значення CPU load

Набір вагових коефіцієнтів	Сценарій	Середнє завантаження CPU (%)
Balanced	Adaptive Full Battery	49.65
	Adaptive Low Battery	31.06
Security	Adaptive Full Battery	51.29
	Adaptive Low Battery	31.42
Energy	Adaptive Full Battery	51.54
	Adaptive Low Battery	31.34

Адаптивні сценарії демонструють гнучку реакцію на зміну умов середовища, що наочно відображено на графіках завантаження процесора. У сценарії Adaptive Full Battery спостерігають динамічну зміну показників синхронно з перемиканням профілів безпеки. У фазах підвищеного рівня загрози (Threat Level) нечіткий логічний контролер (FLC) активує ресурсомісткий профіль Max. Це спричиняє прогнозоване зростання середньої затримки та збільшення навантаження на CPU до рівня, характерного для пікового оброблення (рис. 4.7). Згідно з даними табл. 4.6, середній показник завантаження процесора за експеримент становив 49,65%. Після нормалізації загрозової обстановки система автоматично повертається до профілю Balanced, що спричиняє швидке зменшення latency та розвантаження процесора.

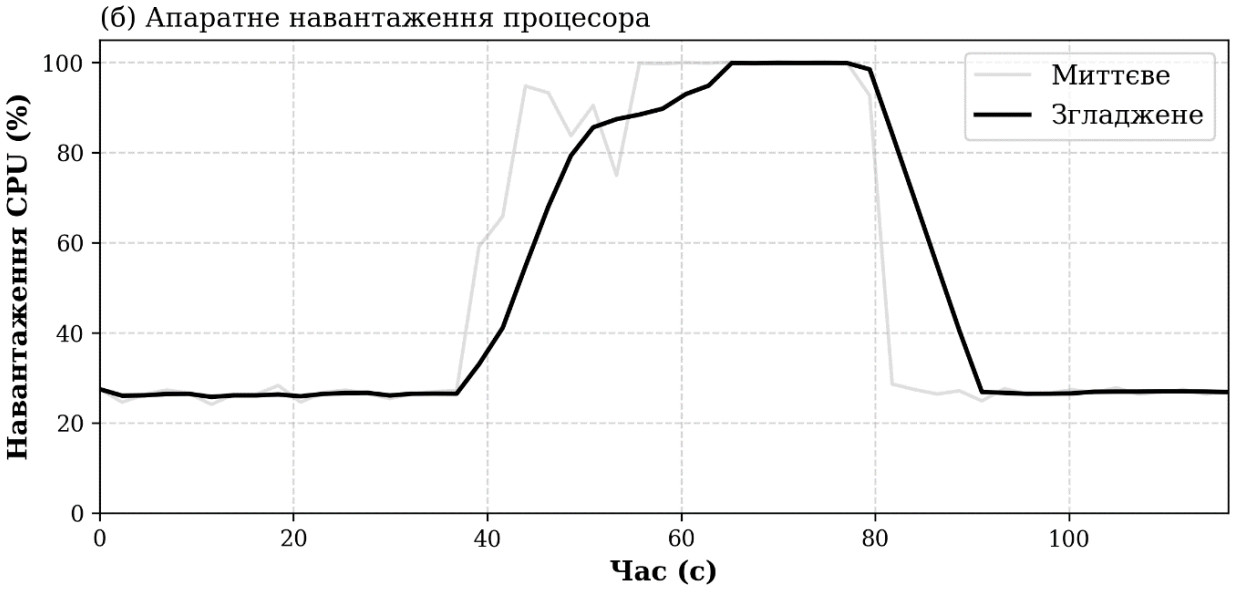

Рисунок 4.7 – Графік навантаження CPU у сценарії Adaptive Full Battery

У сценарії Adaptive Low Battery FLC жорстко обмежує використання ресурсомістких профілів через критично низький рівень заряду батареї. Утримання профілю Survival дає змогу мінімізувати середнє навантаження CPU до 31,06% (рис. 4.8). Таке обмеження формує можливість не лише заощаджувати енергію, а й утримувати середню затримку оброблення в прийнятних межах (до граничних 1000 мс), уникаючи переповнення буферів і відмов в обслуговуванні. Подібні негативні наслідки були б неминучими в разі спроби утримання профілю Max на виснаженій батареї.

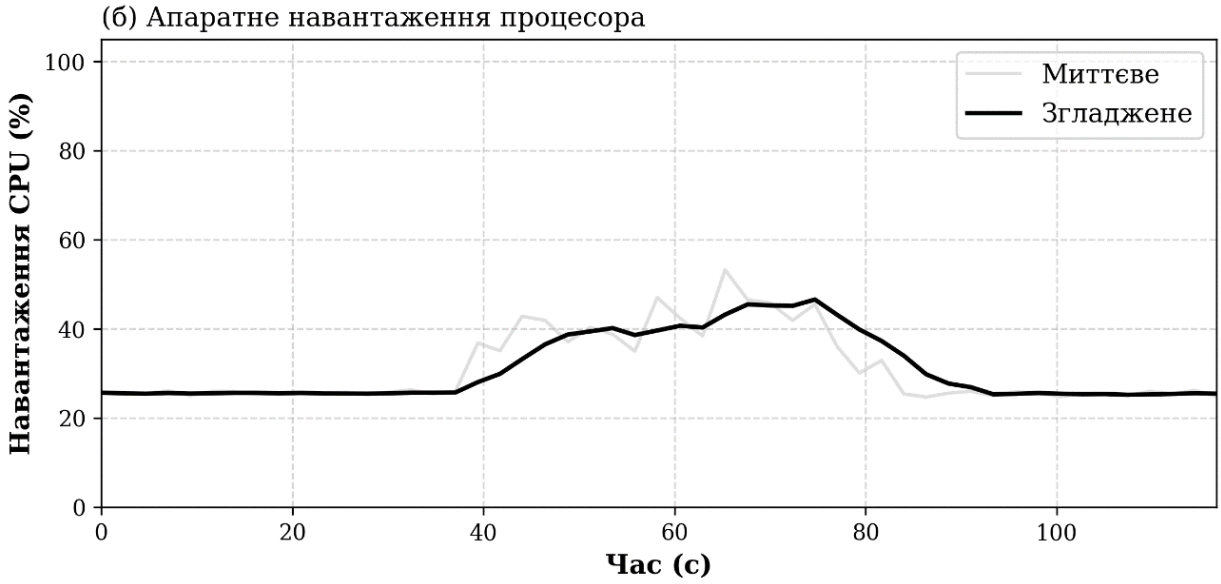

Рисунок 4.8 – Графік навантаження CPU у сценарії Adaptive Low Battery

4.3 Порівняння зі статичними стратегіями

Порівняння адаптивного енергоорієнтованого керування криптографічним захистом fog-вузла з фіксованими криптопрофілями ґрунтують на результатах, які отримано із застосуванням реалізованого ASM-фреймворку на платформі Raspberry Pi 5. Для об’єктивного зіставлення використано метрики енергоспоживання, затримки оброблення, узагальнений показник рівня захисту (Security Score) та інтегральний критерій якості J. Основними об’єктами порівняння є чотири експериментальні стратегії: Adaptive Full Battery, Adaptive Low Battery, Fixed Max та Fixed Balanced. Усі вони досліджені за ідентичних сценаріїв навантаження HTTP-трафіком із DDoS-подібними піками та однаковими профілями зміни рівня загрози (Threat Level).

4.3.1 Постановка задачі порівняння та базові стратегії

Задачу порівняння формулюють як кількісне оцінювання переваг та недоліків адаптивного вибору криптопрофілю, яким керує нечіткий регулятор, відносно статичних стратегій із фіксованим профілем. Оцінювання здійснюють у просторі параметрів «енергія – затримка – безпека», які для зручності багатокритеріального

аналізу зводять до інтегрального критерію якості J. Варто наголосити, що критерій J використовують виключно як аналітичний інструмент для ретроспективного оцінювання ефективності компромісу, а не як цільовий критерій, за якою контролер здійснює безпосередній вибір під час роботи.

Для забезпечення репрезентативності та уніфікації аналізу всі стратегії виконують за однакових сценаріїв: «Full Battery» із початковим станом заряду батареї $B_0 \approx 100\%$ та «Low Battery» із початковим зарядом $B_0 \approx 35\%$. В усіх експериментах використовують спільний профіль інтенсивності запитів та відповідну динаміку зміни рівня загрози (LOW/MEDIUM/HIGH).

Оскільки сучасні процесори класу ARM Cortex-A76 (Raspberry Pi 5) мають апаратну підтримку криптографічних інструкцій, пряме вимірювання енергетичних витрат на оброблення одиничних пакетів є складним завданням. Тому в HiL-стенді застосовано модель синтетичного навантаження, яка відтворює пропорційну обчислювальну складність різних криптопримітивів із застосуванням контрольованих циклів хешування, масштабуючи мікросекундні операції до вимірюваних мілісекундних інтервалів.

Стратегію Fixed Max визначають як політику перманентного застосування профілю Max. У межах стенду цей профіль формує пікове синтетичне навантаження (еквівалент 25 000 ітерацій хешування), що імітує обчислювальну вартість застосування алгоритму AES-256-GCM, доповненого HMAC-SHA-512, незалежно від поточного заряду батареї або оцінки загроз. У цьому режимі fog-вузол постійно функціонує з максимальною обчислювальною складністю. Це гарантує найвищий показник Security Score, проте спричиняє максимальне зростання навантаження на процесор та питомого енергоспоживання порівняно з іншими профілями.

Стратегія Fixed Balanced реалізує незмінний вибір профілю Balanced. Цей режим генерує пропорційно менше навантаження (еквівалент 15 000 ітерацій), що відповідає обчислювальній складності легших криптопримітивів (класу AES-128-GCM у поєднанні з HMAC-SHA-256). Коефіцієнт навантаження між профілями Max та Balanced становить 1.66, що зберігає об'єктивні пропорції їхньої ресурсомісткості. Цей режим формує можливість стабілізувати затримки оброблення та

енергоспоживання на проміжному рівні. Проте статичність цієї політики унеможливлює реакцію на зміну Threat Level і не забезпечує раціонального використання енергії акумулятора в енергообмежених IoT-сценаріях.

Стратегії типу Fixed Survival аналізують як нижню межу ресурсоорієнтованої політики, де синтетичне навантаження знижено до мінімуму (еквівалент 6 000 ітерацій), що відтворює роботу легкого потокового шифру ChaCha20 із помірною цілісністю. Такі режими кардинально зменшують навантаження на процесор та енергоспоживання, але не здатні забезпечити високу криптографічну стійкість, критично необхідну для fog-вузлів під час атак. У межах проведених ASM-експериментів досліджено дві адаптивні конфігурації. Сценарій Adaptive Full Battery ($B_0 \approx 100\%$) орієнтовано на досягнення максимальної безпеки за умови наявності достатнього енергоресурсу. Сценарій Adaptive Low Battery ($B_0 \approx 35\%$) має на меті забезпечення живучості fog-вузла, надаючи пріоритет менш ресурсомістким профілям в умовах початкового критичного дефіциту енергії.

4.3.2 Порівняння енергетичної ефективності

Енергетичну ефективність оцінюють за швидкістю розряджання модельованої батареї та часом досягнення критичного порога рівня заряду. В експериментах на базі ASM-стенду для fog-вузла Raspberry Pi 5 цей поріг визначає рівень, нижче за який вузол вважають непридатним до надійного оброблення трафіку. Оскільки пряме апаратне вимірювання споживаного струму під час короткочасних експериментів є ускладненим, програмна модель батареї використовує зв’язок між завантаженням процесора (CPU load) та емпіричними параметрами розряду (струм у режимах idle та max). Це створює можливість об’єктивно співвіднести синтетичну обчислювальну складність поточного криптопрофілю з динамікою енергетичних втрат.

Рівень розряду для всіх сценаріїв за використання трьох наборів ваг подано в табл. 4.7.

Таблиця 4.7 – Рівень розряду для всіх сценаріїв за трьох наборів ваг

Набір вагових коефіцієнтів	Сценарій	Рівень розряду (%)
Balanced	Adaptive Full Battery	15.527
	Adaptive Low Battery	10.039
	Fixed Max	16.71
	Fixed Balanced	12.096
Security	Adaptive Full Battery	15.381
	Adaptive Low Battery	9.945
	Fixed Max	15.862
	Fixed Balanced	11.671
Energy	Adaptive Full Battery	14.816
	Adaptive Low Battery	9.514
	Fixed Max	15.811
	Fixed Balanced	11.451

Аналіз базових статичних стратегій підтверджує їхню нераціональність в умовах дефіциту ресурсів. Зокрема, сценарій Fixed Max характеризується максимальною швидкістю розряду: частка витраченої енергії (EnergyDelta) сягає 16,71% за цикл експерименту. Це пояснюють тим, що кожен запит безальтернативно обробляють із використанням імітації найважчих криптопримітивів (AES-256-GCM та HMAC-SHA-512). Як наочно відображає рис. 4.9, такий підхід генерує максимальне завантаження процесора і, відповідно, спричиняє найшвидше падіння рівня заряду.

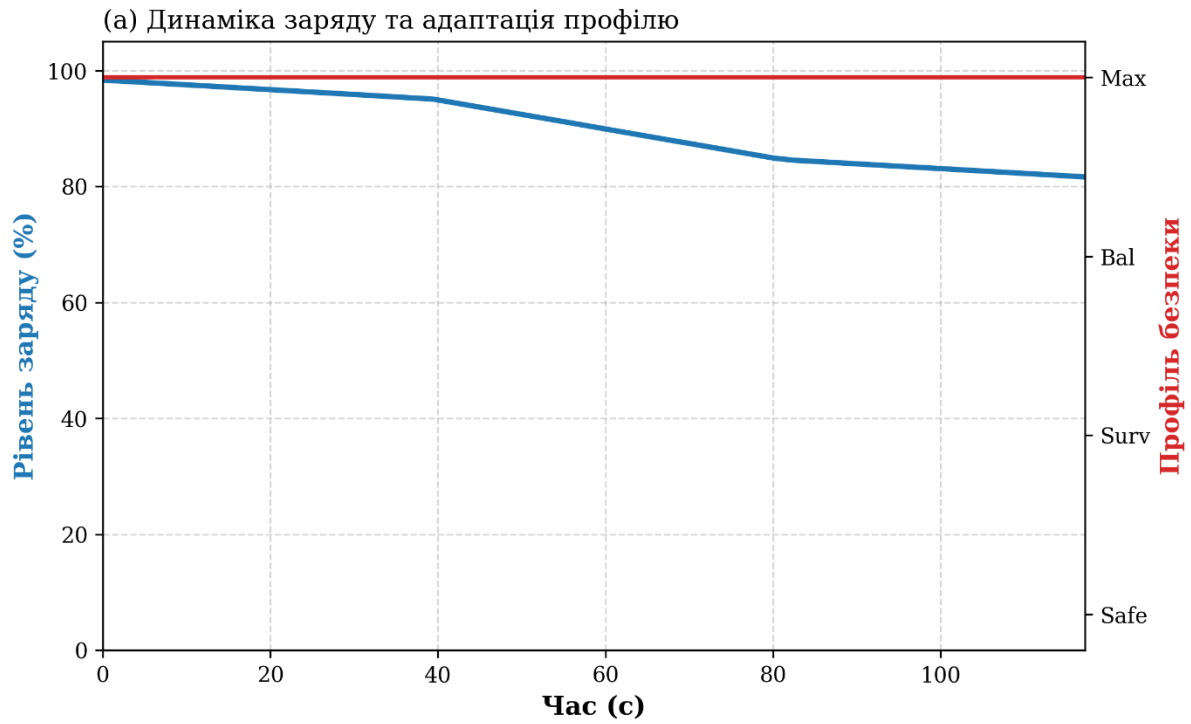

Рисунок 4.9 – Графік енергоспоживання сценарію Fixed Max

У сценарії Fixed Balanced постійне застосування алгоритмів середньої складності (AES-128-GCM та HMAC-SHA-256) забезпечує зниження показника EnergyDelta до рівня 12,10% (рис. 4.10). Проте відсутність адаптації зумовлює те, що в умовах низького рівня загроз обчислювальний ресурс витрачають нераціонально, оскільки вузол не переходить до енергоощадних профілів, а під час пікових атак (Threat Level = HIGH) система виявляється неспроможною підвищити рівень захисту до необхідного максимуму.

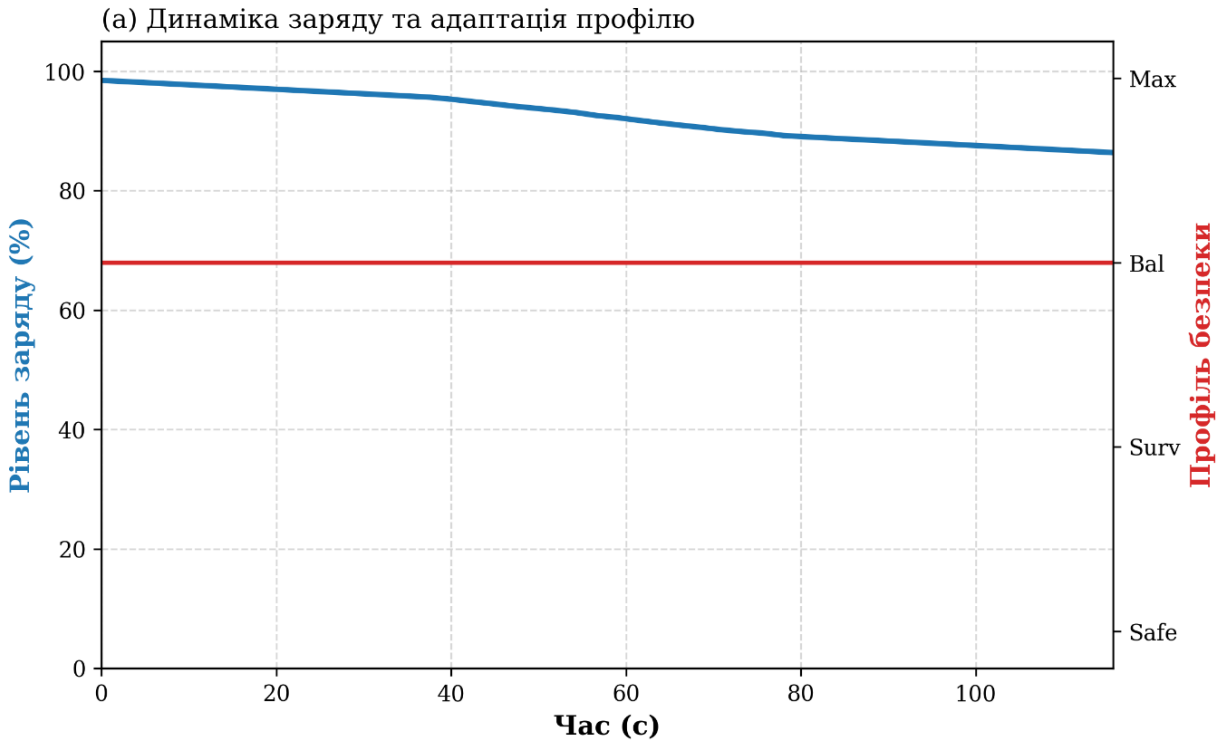

Рисунок 4.10 – Графік енергоспоживання сценарію Fixed Balanced

Адаптивна стратегія Adaptive Full Battery в умовах повного заряду батареї демонструє енергоспоживання, близьке до Fixed Max, лише під час фаз високого Threat Level, коли нечіткий логічний контролер (FLC) виправдано активує ресурсомісткий профіль Max. Однак загальна витрата енергії (EnergyDelta) залишається стабільно меншою, ніж у Fixed Max (у середньому 15,53%, згідно з табл. 4.7). Цю економію досягають внаслідок автоматичного переходу до профілів Balanced або Survival у разі зниження загрозового фону. Важливо відзначити, що це фактичне зменшення енерговитрат безпосередньо зумовлює меншу величину нормалізованого компонента P_{norm}. Відповідно, під час подальшого математичного оцінювання із використанням інтегрального критерію якості J (особливо з енергоорієнтованим набором ваг), адаптивна стратегія отримує значно менший штраф за енергоспоживання зі збереженням високого середнього показника безпеки.

Найбільший енергетичний виграш забезпечує стратегія Adaptive Low Battery. У цьому сценарії початковий критичний рівень заряду (близько 35%) є ключовим фактором для FLC, який примусово обмежує систему легкими профілями Survival і

Balanced навіть в умовах підвищеного Threat Level. Зниження EnergyDelta до рівня близько 10,0% суттєво відтерміновує досягнення критичного заряду батареї порівняно зі стратегією Fixed Max. За ідентичного профілю мережевого навантаження Adaptive Low Battery забезпечує збільшення тривалості автономної роботи майже в 1,7 раза, успішно реалізуючи режим керованої деградації, за якого fog-вузол зберігає операційну працездатність ціною допустимого зниження криптографічної стійкості.

4.3.3 Порівняння навантаження CPU

У сценарії Fixed Max затримка оброблення (latency) та навантаження на центральний процесор (CPU load) очікувано залишаються стабільно найвищими протягом усього експерименту. Оскільки кожен запит, незалежно від контексту та поточного рівня загрози, безальтернативно обробляють із використанням найбільш ресурсомісткого профілю Max (синтетична імітація криптографічних операцій класу AES-256-GCM та HMAC-SHA-512), середнє навантаження процесора сягає 53,69% (рис. 4.11). Це зумовлює максимальну тривалість обчислювальних перетворень і призводить до потенційного формування черг на оброблення навіть за умов фонового трафіку та відсутності кібератак.

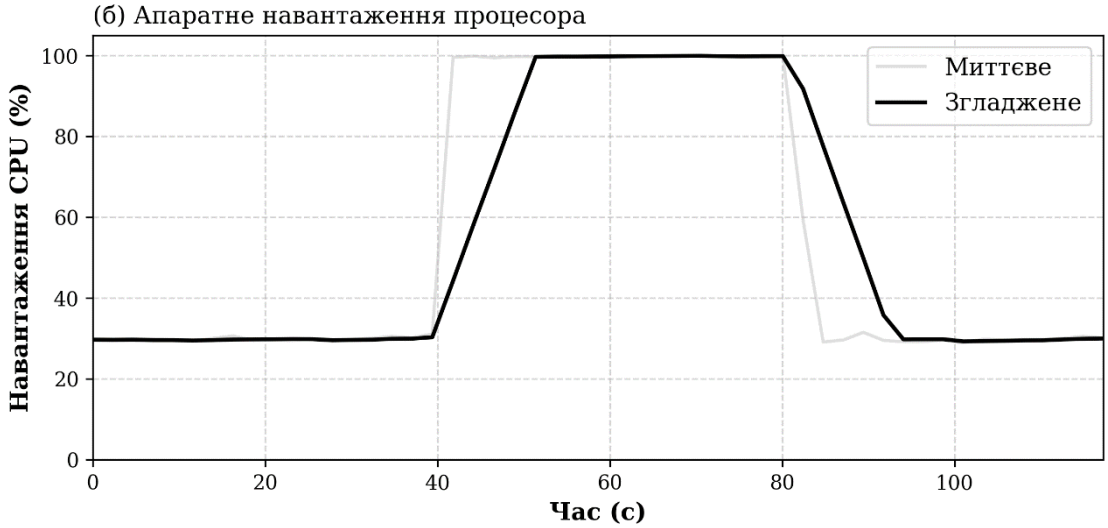

Рисунок 4.11 – Графік навантаження CPU у сценарії Fixed Max

Сценарій Fixed Balanced демонструє нижчі середні значення завантаження CPU – на рівні 38,86% (рис. 4.12). Це досягають внаслідок перманентного застосування пропорційно полегшеного синтетичного навантаження, що еквівалентне криптопримітивам класу AES-128-GCM та HMAC-SHA-256. Однак статичність цієї політики унеможливлює адекватну реакцію системи на зміну показника Threat Level. Унаслідок цього обчислювальний ресурс витрачають нераціонально під час фаз низької мережевої активності, а під час пікових DDoS-подібних аномалій система фізично не здатна перейти на вищий рівень захисту, наражаючи fog-вузол на небезпеку компрометації.

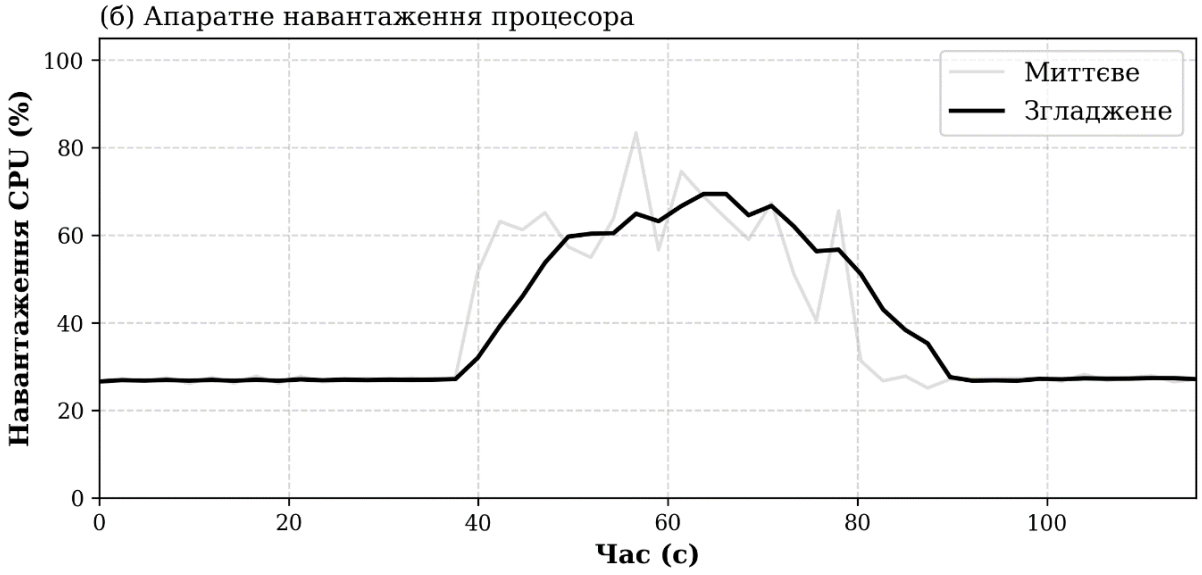

Рисунок 4.12 – Графік навантаження CPU у сценарії Fixed Balanced

Отже, реалізована в ASM-фреймворку адаптивна політика керування (порівняльні результати якої зведено в табл. 4.8) підтверджує здатність системи ефективно балансувати між затримкою оброблення даних і рівнем криптографічного захисту. Динамічний вибір профілів із застосуванням нечіткого логічного контролера (FLC) суттєво зменшує середню затримку оброблення порівняно зі сценарієм Fixed Max та забезпечує значно вищу реактивність щодо загроз порівняно з Fixed Balanced.

Таблиця 4.8 – Середні значення CPU load

Набір вагових коефіцієнтів	Сценарій	Середнє завантаження CPU (%)
Balanced	Adaptive Full Battery	49.65
	Adaptive Low Battery	31.06
	Fixed Max	53.69
	Fixed Balanced	38.86
Security	Adaptive Full Battery	51.29
	Adaptive Low Battery	31.42
	Fixed Max	53.04
	Fixed Balanced	38.23
Energy	Adaptive Full Battery	51.54
	Adaptive Low Battery	31.34
	Fixed Max	55.25
	Fixed Balanced	39.21

Особливо показовим у контексті раціонального використання ресурсів є адаптивний сценарій за умов низького заряду (Adaptive Low Battery). Застосована в ньому стратегія керованої деградації гарантує уникнення процесорного перевантаження на етапі критичного дефіциту енергії та стабілізує середнє навантаження CPU на безпечному рівні 31,06%. Такий підхід зберігає обчислювальну стабільність, мінімізує ризики відмов в обслуговуванні через переповнення буферів та забезпечує керованість fog-вузла до моменту повного вичерпання ресурсу батареї.

4.3.4 Порівняння затримки оброблення запитів

Якість наданого сервісу оцінюють через середню затримку оброблення HTTP-запитів fog-вузлом. На базі реалізованого ASM-стенду затримку фіксують як часовий інтервал між надходженням запиту до API fog-вузла та генерацією відповіді клієнту. Цей показник інтегрує в собі час виконання імітаційного криптографічного

навантаження, загальні витрати процесорного часу та затримки на взаємодію з внутрішніми контейнеризованими сервісами.

Стратегія Fixed Max очікувано демонструє найвищу середню затримку оброблення серед усіх розглянутих конфігурацій (табл. 4.9). Це є прямим наслідком перманентного застосування найбільш ресурсомісткого профілю (еквівалент 25 000 ітерацій гешування для імітації AES-256-GCM та HMAC-SHA-512) під час обслуговування кожного вхідного запиту. Відповідно, середня затримка Fixed Max об’єктивно перевищує аналогічний показник адаптивної стратегії Adaptive Full Battery за ідентичного профілю інтенсивності мережевого навантаження (RPS). Це яскраво свідчить про нераціональне використання обчислювальних ресурсів системою зі статичною політикою керування.

Таблиця 4.9 – Середня затримка для всіх сценаріїв за трьох наборів ваг

Набір вагових коефіцієнтів	Сценарій	Середня затримка (мс)
Balanced	Adaptive Full Battery	19,638
	Adaptive Low Battery	5,898
	Fixed Max	32,128
	Fixed Balanced	14,521
Security	Adaptive Full Battery	20,189
	Adaptive Low Battery	5,827
	Fixed Max	31,912
	Fixed Balanced	14,072
Energy	Adaptive Full Battery	19,813
	Adaptive Low Battery	5,738
	Fixed Max	32,127
	Fixed Balanced	13,992

Сценарій Fixed Balanced демонструє стабільно нижчу затримку порівняно з Fixed Max. Це зумовлено помітно меншою обчислювальною складністю профілю Balanced (еквівалент 15 000 ітерацій), що імітує застосування криптопримітивів класу AES-128-GCM та HMAC-SHA-256 і, відповідно, потребує меншої кількості тактів

процесора ARM-ядра Raspberry Pi 5. Проте фундаментальним недоліком цієї статичної стратегії є неможливість підвищення рівня захисту. Навіть у фазах високого рівня загрози (Threat Level = HIGH), коли агресивні DDoS-подібні аномалії вимагають максимальної стійкості механізмів автентифікації й шифрування, система залишається вразливою.

Натомість адаптивні стратегії керують показником затримки динамічно. У фазах високого рівня загрози нечіткий логічний контролер (FLC) підвищує активний Security Profile до рівня Max. Це прогнозовано збільшує затримку до значень, характерних для Fixed Max, проте такі інтервали є обмеженими в часі й не домінують у загальній тривалості роботи системи. Під час нормалізації загрозового фону або в разі падіння рівня заряду батареї (зокрема, у сценарії Adaptive Low Battery), FLC переводить систему до легших профілів Balanced або Survival. Це суттєво знижує середню затримку та дає змогу уникати довготривалих піків переповнення черг оброблення без критичного зниження рівня безпеки нижче від допустимої межі.

Вбудовані механізми запобігання брязкоту (зокрема, порогова логіка відображення безперервної оцінки Security Score на дискретні профілі) в ASM-агенті ефективно обмежують частоту перемикань. Це запобігає виникненню швидких циклічних осциляцій між режимами Max, Balanced та Survival в умовах незначних флуктуацій рівня заряду батареї або фону загроз поблизу порогових значень. Експериментальні журнали підтверджують, що адаптивні стратегії не вносять додаткової нестабільності в обслуговування трафіку: дисперсія затримок залишається в прогнозованих межах, формуючи базову лінію на рівні профілю Balanced із локальними обґрунтованими підвищеннями виключно у фазах високого рівня загрози.

4.3.5 Порівняння рівня безпеки

Рівень безпеки в ASM-фреймворку оцінюють через середнє значення узагальненого показника Security Score та мінімальний досягнутий рівень криптографічного профілю протягом експерименту. Security Score математично інтегрує частку часу, яку система проводить у профілях Safe, Survival, Balanced і Max.

Кожному профілю зіставлено фіксовану вагу (відповідно 10, 30, 60 та 90), що формує можливість кількісно формалізувати компромісний зв’язок між енергозатратністю застосованих криптографічних імітацій та забезпеченим рівнем конфіденційності, цілісності й автентичності. Зведені результати середнього показника безпеки під час експериментів подано в табл. 4.10.

Таблиця 4.10 – Середнє значення Security Score за різних наборів ваг J

Набір вагових коефіцієнтів	Сценарій	Середній показник безпеки
Balanced	Adaptive Full Battery	70.0
	Adaptive Low Battery	40.1
	Fixed Max	90.0
	Fixed Balanced	60.0
Security	Adaptive Full Battery	70.4
	Adaptive Low Battery	40.2
	Fixed Max	90.0
	Fixed Balanced	60.0
Energy	Adaptive Full Battery	70.2
	Adaptive Low Battery	40.4
	Fixed Max	90.0
	Fixed Balanced	60.0

Стратегія Fixed Max очікувано забезпечує максимальний середній показник Security Score (на рівні 90), оскільки fog-вузол безальтернативно перебуває в профілі Max. Ця стратегія гарантує повну відсутність зниження профілю до рівня Balanced чи Survival, що формально відповідає максимально можливому рівню захисту в межах запропонованого фреймворку. Проте цей результат досягають ціною критичного підвищення енергоспоживання та затримок, що у сценаріях із дефіцитом заряду (Low Battery) спричиняє швидку і неминучу втрату живучості вузла, знецінюючи високий показник безпеки глобальною відмовою системи.

Стратегія Fixed Balanced забезпечує стабільний компромісний рівень захисту із середнім значенням Security Score поблизу 60. Цього показника категорично

недостатньо у фазах високого рівня загрози (Threat Level = HIGH), коли агресивний атакувальний трафік вимагає максимальної криптографічної стійкості. Через відсутність механізмів динамічного підвищення профілю до рівня Max, стратегія Fixed Balanced ігнорує реальне зростання ризиків компрометації, що є прямим порушенням вимог до контекстно-залежного керування захистом у fog-середовищах.

Адаптивна стратегія Adaptive Full Battery демонструє високий середній Security Score, який наближається до показників Fixed Max. Це досягають внаслідок тривалого перебування вузла у профілі Max під час імітації DDoS-атак та раціонального використання Balanced у менш критичних фазах. Під час подальшого математичного оцінювання цих результатів із застосуванням інтегрального критерію якості J у разі використання жорстко безпеково-орієнтованого набору ваг (де втрати енергії ігнорують) Fixed Max закономірно залишається математичним референтом мінімального штрафу, проте лише за умови, що експеримент завершується до моменту повного розряджання батареї.

У сценарії Adaptive Low Battery нечіткий логічний контролер (FLC) вимушено застосовує профілі Survival і Balanced частіше, щоб утримати Security Score на мінімально допустимому рівні, який визначає політика безпеки. Під час роботи з профілем Survival ASM-фреймворк не допускає зниження профілю нижче від Safe. Це гарантує збереження базової цілісності трафіку (імітація CRC32) та підтримання мінімального рівня конфіденційності (імітація ChaCha20 у профілі Survival) для критичних ділянок. Такий підхід ідеально відповідає концепції керованої деградації у survivability-орієнтованих архітектурах, оскільки створює умови для fog-вузла функціонувати під навантаженням значно довше, ніж за умови статичної політики Fixed Max у разі аналогічного критичного стартового рівня заряду батареї.

4.3.6 Аналіз компромісу через інтегральний критерій J

З метою багатокритеріального оцінювання ефективності запропонованого методу та кількісного аналізу компромісів між затримкою, енергоспоживанням і рівнем безпеки застосовано інтегральний критерій якості J, математичну модель якого

обґрунтовано в розділі 2 (2.19). Розрахунок критерію для всіх досліджених стратегій за різних наборів вагових коефіцієнтів (збалансованих, безпеково- та енергоорієнтованих) дає змогу математично позиціонувати адаптивні підходи відносно статичних профілів. Узагальнення розрахованих значень інтегрального критерію якості J для всіх сценаріїв наведено в табл. 4.11.

Таблиця 4.11 – Узагальнення значень J для всіх сценаріїв за трьох наборів ваг

Набір вагових коефіцієнтів	Сценарій	Середнє значення J
Balanced	Adaptive Full Battery	0.238
	Adaptive Low Battery	0.549
	Fixed Max	0.2085
	Fixed Balanced	0.2486
Security	Adaptive Full Battery	0.2674
	Adaptive Low Battery	0.5731
	Fixed Max	0.1529
	Fixed Balanced	0.3229
Energy	Adaptive Full Battery	0.1617
	Adaptive Low Battery	0.6295
	Fixed Max	0.1505
	Fixed Balanced	0.1592

Аналіз динаміки інтегральної метрики J засвідчує, що її значення відображають не стільки абсолютну перевагу одного з режимів у вакуумі, скільки специфіку заданих вагових пріоритетів на обмеженому часовому інтервалі спостереження.

Зокрема, за збалансованого набору ваг статичний сценарій Fixed Max демонструє формально нижче значення штрафу ($J = 0{,}2085$) порівняно з адаптивною стратегією Adaptive Full Battery ($J = 0{,}2380$). З математичного погляду це пояснюють вагомим внеском безпекової складової: постійне застосування максимального криптографічного профілю зводить до мінімуму нормалізований штраф за безпеку, що на короткому проміжку часу перекриває втрати від підвищеного енергоспоживання. Проте екстраполяція цих даних показує, що така перевага є

тимчасовою. Стратегія Fixed Max неминуче спричинить швидке вичерпання ресурсу батареї (що еквівалентно відмові вузла, $J \to 1$), тоді як Adaptive Full Battery забезпечує стабільний компроміс, наближаючись до референтних значень безпеки без ризику передчасного відключення.

В умовах безпеково-орієнтованого набору ваг (Security), де енергетичні втрати мінімально впливають на критерій, Fixed Max закономірно виступає математичним еталоном із найнижчим значенням $J = 0{,}1529$. Важливо відзначити, що стратегія Adaptive Full Battery наближається до цього референтного значення ($J = 0{,}2674$). Це є ключовим показником ефективності нечіткого регулятора: система демонструє лише відхилення від ідеального рівня безпеки, водночас забезпечуючи об'єктивно менше фізичне енергоспоживання (EnergyDelta) у фазах фонового трафіку.

Найбільш показовою є оцінка поведінки системи в умовах дефіциту ресурсу (сценарій Adaptive Low Battery). За умови використання енергоорієнтованих ваг (Energy), для цієї стратегії фіксують відносно високе значення інтегрального штрафу ($J = 0{,}6295$). Такий математичний результат зумовлений значним штрафом за вимушено низький рівень захищеності (середній показник Security Score становить 40,4 через домінування профілю Survival).

Однак цей високий штраф є закономірним наслідком об'єктивних фізичних обмежень, а не недоліком алгоритму. В умовах критичного стартового заряду (близько 35%) нечіткий логічний контролер ініціює керовану деградацію виключно на основі зниження заряду батареї, щоб запобігти відмові вузла. Хоча лінійний критерій J фіксує це як зростання штрафу за безпеку, на практиці такий крок є єдиним виправданим інженерним рішенням. Завдяки суттєвому зниженню обчислювального навантаження на процесор (до 31,06%) та зменшенню енергетичних витрат (EnergyDelta $\approx 10{,}04\%$), fog-вузол здатний подовжити час автономної роботи під навантаженням у 1,7 раза порівняно зі статичною стратегією.

Отже, результати комплексного аналізу (табл. 4.12) доводять, що ефективність адаптивного керування на базі контролера з нечіткою логікою полягає у здатності системи динамічно балансувати ресурсами. За наявності достатнього енергетичного ресурсу система мінімізує інтегральний штраф J, наближаючись до еталонного

захисту. Натомість за умов критичного дефіциту заряду батареї пріоритетом стає фізичне виживання fog-вузла: контролер ігнорує спроби утримання в високих значень функції штрафу задля збереження глобальної живучості системи, що робить цей підхід найбільш доцільним для автономних IoT-інфраструктур.

Таблиця 4.12 – Зведені результати порівняльного аналізу

Набір вагових коефіцієнтів	Сценарій	Середнє значення J	Розряд (%)	Середнє значення CPU (%)	Середній показник безпеки
Balanced	Adaptive Full Battery	0.238	15.527	49.65	70.0
	Adaptive Low Battery	0.549	10.039	31.06	40.1
	Fixed Max	0.2085	16.71	53.69	90.0
	Fixed Balanced	0.2486	12.096	38.86	60.0
Security	Adaptive Full Battery	0.2674	15.381	51.29	70.4
	Adaptive Low Battery	0.5731	9.945	31.42	40.2
	Fixed Max	0.1529	15.862	53.04	90.0
	Fixed Balanced	0.3229	11.671	38.23	60.0
Energy	Adaptive Full Battery	0.1617	14.816	51.54	70.2
	Adaptive Low Battery	0.6295	9.514	31.34	40.4
	Fixed Max	0.1505	15.811	55.25	90.0
	Fixed Balanced	0.1592	11.451	39.21	60.0

Результати експериментальних досліджень підтверджують, що за умов критичного енергодефіциту або інтенсивної навантажувальної атаки утримання максимального профілю захисту призводить до стрімкого знеструмлення електронної платформи. У такій ситуації апаратний Fog-вузол повністю втрачає працездатність, що означає нульовий рівень захищеності внаслідок відмови в обслуговуванні. Застосування профілю Survival дає змогу суттєво зменшити струми споживання

мікропроцесорного ядра, зберігаючи конфіденційність і цілісність даних. Такий підхід забезпечує виграш часу автономної роботи, необхідний для безпечного завершення критичних процесів або відновлення стабільного енергопостачання, що є пріоритетним завданням для автономних систем електроніки.

4.4 Робастність, чутливість, обмеження та рекомендації впровадження

Запропонований енергоорієнтований адаптивний механізм захисту в ASM-фреймворку забезпечує гнучкий компроміс між енергоспоживанням, затримкою оброблення та рівнем захисту в широкому діапазоні умов. Поведінку системи під час роботи визначають виключно налаштування контролера нечіткої логіки (FLC), тоді як результати багатокритеріального оцінювання цієї поведінки залежать від вибору вагових коефіцієнтів інтегрального критерію якості J. Результати апаратно-програмних експериментів на HiL-стенді демонструють прогнозовану зміну за критерієм J у разі варіації аналітичних політик (Balanced, Pro-Security, Pro-Energy), до того ж фізична логіка вибору криптопрофілів системою залишається стабільною.

Робастність запропонованого механізму визначають як його здатність підтримувати ефективне функціонування fog-вузла та забезпечувати адекватний рівень криптографічного захисту в умовах непередбачуваних варіацій характеристик зовнішнього середовища (профіль мережевого трафіку, інтенсивність кібератак, початковий рівень заряду акумулятора). Для комплексного дослідження меж застосовності розробленого методу, задачу аналізу чутливості необхідно розділити на дві незалежні складові, що відповідають архітектурі рішення:

1) Аналіз чутливості математичної оцінки: дослідження впливу зміни вагових коефіцієнтів (w_L, w_P, w_S) у формулі інтегрального штрафу (2.19) на кінцеве значення J для різних сценаріїв без зміни їхньої фізичної поведінки.
2) Аналіз чутливості алгоритму керування: дослідження впливу зміни внутрішніх параметрів регулятора (параметрів функцій належності FLC для Battery Level та Threat Level, а також порогів дискретизації Security Score)

безпосередньо на фізичну траєкторію розряду батареї та часовий розподіл обраних криптопрофілів (Safe, Survival, Balanced, Max).

Таке чітке розмежування математичного оцінювання та фізичного керування є ключовим для практичної адаптації ASM-фреймворку під різні класи застосувань. Зміна параметрів бази знань FLC формує можливість для інженерів фізично підлаштувати логіку вузла під конкретні умови (від енергообмежених автономних дронів до стаціонарних fog-шлюзів), тоді як варіація ваг критерію J слугує аналітичним інструментом для математичної валідації успішності цієї адаптації.

4.4.1 Чутливість до вагових коефіцієнтів інтегрального критерію J

У реалізованому ASM-фреймворку для багатокритеріального аналізу запропоновано три типові набори вагових коефіцієнтів інтегрального критерію J, які відповідають різним експлуатаційним пріоритетам: Balanced, Pro-Security та Pro-Energy. Набір Balanced реалізований як $w_L = 0.33, w_P = 0.33, w_S = 0.34$, що забезпечує симетричний вплив складових. Конфігурація Pro-Security надає домінувальну вагу безпеці (наприклад, $w_S = 0.7$) за умови знижених $w_L = 0.2$ та $w_P = 0.1$, тоді як Pro-Energy пропорційно підсилює значущість енергетичної складової ($w_P = 0.6$). Такий розподіл математично відтворює практику формулювання вимог до QoS у різнотипових завданнях Інтернету речей.

Математичний аналіз результатів демонструє, як зміна вагових коефіцієнтів впливає на отримане значення J. За збалансованих ваг (Balanced) найкращий результат показує статична стратегія Fixed Max, хоча адаптивна стратегія Adaptive Full Battery демонструє близьке та конкурентне значення.

У разі переходу до набору ваг Pro-Security глобальний мінімум критерію закономірно зміщується в бік Fixed Max. Це пояснюється тим, що високий показник безпеки у формулі цільового критерію повністю компенсує енергетичні втрати та зростання затримки.

Для набору ваг Pro-Energy результати аналізу виявляються дещо специфічними: попри пріоритет енергозбереження, мінімальне значення штрафу знову досягається

для стратегії Fixed Max. Водночас стратегія Adaptive Low Battery, попри закладені в нечіткий логічний контролер (FLC) профілі енергозбереження (Survival та Balanced), отримує найбільший математичний штраф у всій таблиці. У цій конфігурації статична стратегія Fixed Max виявляється аналітично ефективнішою, оскільки навіть при домінуванні енергетичної ваги, критичне падіння показника безпеки в адаптивних сценаріях не здатне компенсувати виграш від економії заряду.

Отримана математична суперечність пояснюється лінійною природою критерію J та різним масштабом вхідних параметрів, де значне падіння Security Score (з 90.0 до 40.4) формально переважає над фізичним збереженням енергії. Проте з позиції електроніки та системного проєктування Fog-мереж, така аналітична перевага статичної стратегії є ілюзорною. В умовах низького заряду батареї критичним показником стає апаратна живучість вузла. Повне розрядження акумулятора в стратегії Fixed Max призводить до вимкнення пристрою, що фактично обнуляє показник безпеки та робить вузол недоступним. Таким чином, адаптивна стратегія забезпечує фізичне функціонування системи в екстремальних умовах, що є пріоритетним над умовним мінімумом математичного штрафу.

Фундаментально важливо наголосити на архітектурній розв'язці запропонованого фреймворку: зміна вагових коефіцієнтів у критерії J жодним чином не впливає на фізичну поведінку fog-вузла або логіку ухвалення рішень контролером FLC. Як підтверджують часові діаграми (рис. 4.5–4.8), алгоритм роботи системи, зокрема інтенсивність енергозбереження, визначають виключно об'єктивні вхідні дані – поточний рівень заряду батареї та рівень кіберзагрози (Threat Level). Інтегральний критерій якості J є ізольованим аналітичним інструментом ретроспективної валідації. Тому перерозподіл лідерства між адаптивними та статичними стратегіями за критерієм J у разі зміни ваг ілюструє не адаптацію самої системи, а виключно зміну координат оптимуму в просторі оцінювання.

Відповідно, наведені результати формують чіткі рекомендації щодо аналітичного супроводу ASM-фреймворку. Вибір набору ваг варто використовувати для цільової оцінки (валідації) відповідності системи вимогам конкретного сервісу: ефективність роботи безпеково-критичних застосунків доцільно валідувати з

використанням ваг Pro-Security, тоді як життєздатність енергообмежених мобільних вузлів слід перевіряти за вагами Pro-Energy. Для фактичної ж зміни поведінки вузла (його конфігурування) необхідно застосовувати налаштування параметрів бази правил самого FLC, що детально проаналізовано далі.

4.4.2 Чутливість до параметрів нечіткого контролера

Як було доведено в попередньому підрозділі, фактична фізична поведінка fog-вузла не залежить від вагових коефіцієнтів функції оцінювання, а її визначають виключно внутрішні налаштування контролера нечіткої логіки (FLC). У розробленому ASM-фреймворку процес фазифікації оперує трьома лінгвістичними терм-множинами для вхідної змінної Battery Level (LOW, MEDIUM, HIGH) та відповідними рівнями для Threat Level (LOW, MEDIUM, HIGH). Базова конфігурація використовує трапецієподібні та трикутні функції належності, параметри яких зафіксовано емпірично (наприклад, терм LOW для рівня заряду батареї до ≈ 20, MEDIUM у діапазоні $\approx 20 - 60$, HIGH – понад ≈ 60). Вихідний сигнал після дефазифікації формують у вигляді чіткого (crisp) значення Security Score (від 0 до 100) з подальшою дискретизацією до профілів Safe, Survival, Balanced, Max за пороговими значеннями 20, 50 і 80 відповідно.

Зміна меж між зонами Battery Level є прямим механізмом конфігурування енергоефективності системи. Наприклад, зсув порогу активації терму LOW у більший бік (наприклад, з 20% до 35%) примусово ініціює більш ранній перехід системи до енергоощадного профілю Survival. Як наочно демонструє динаміка розряду в сценарії Adaptive Low Battery (рис. 4.6), це зумовлює об'єктивне зниження навантаження на центральний процесор та суттєве зменшення енергетичних витрат (EnergyDelta), що, однак, закономірно зумовлює зниження середнього показника Security Score. У зворотному випадку (заниження порогу LOW) система затримує активацію режиму виживання, розширюючи часовий інтервал застосування ресурсомістких профілів Balanced та Max. Це підвищує криптографічну стійкість під час атак, але критично пришвидшує розряджання батареї.

Форма функцій належності та ступінь перекриття між сусідніми термами (LOW/MEDIUM/HIGH) визначають динаміку перехідних процесів в алгоритмі керування. Розширення зон перекриття забезпечує більш плавні зміни вихідного показника Security Score та виконує функцію вбудованого механізму запобігання брязкоту (anti-chattering). Це знижує ймовірність високочастотних перемикань криптопрофілів в умовах незначних флуктуацій заряду батареї або Threat Level, що цілком відповідає класичним рекомендаціям щодо синтезу нечітких регуляторів. Водночас надмірне розширення зон перекриття здатне спричинити втрату чутливості системи (інерційність) у разі стрімкого падіння заряду через пікове навантаження або за різких стрибків рівня кіберзагроз, що є неприпустимим у безпеково-критичних промислових сценаріях.

Пороги дискретизації вихідної змінної Security Score (20, 50, 80) безпосередньо визначають чутливість системи до активації конкретних криптографічних імітацій. Зниження порогу для профілю Max (наприклад, з 80 до 70) зумовлює частіше застосування пікового навантаження навіть за умов помірних значень Threat Level та достатнього заряду акумулятора. Це гарантує максимальний рівень захисту, проте закономірно спричиняє зростання EnergyDelta та середньої затримки оброблення запитів (latency). З іншого боку, підвищення порогів для профілів Survival і Safe конфігурує систему як більш агресивну в контексті енергозбереження: в умовах логічної невизначеності FLC частіше обиратиме легкі профілі, свідомо знижуючи криптостійкість задля максимізації часу автономної роботи вузла.

Загальний аналіз варіацій параметрів нечіткого виведення засвідчує, що запропонований FLC зберігає стабільну та передбачувану фізичну поведінку в широкому діапазоні конфігурацій. На відміну від зміни вагових коефіцієнтів оціночного критерію якості J, ці налаштування забезпечують реальну адаптацію системи до умов експлуатації. Надто агресивні безпекові конфігурації (занижені пороги для ввімкнення профілю Max або звужені зони розмежування LOW) перетворюють адаптивну систему на квазістатичну, наближаючи її поведінку до сценарію Fixed Max (високий захист, швидкий розряд). Натомість надмірно

ресурсоощадні налаштування наближають поведінку до сценарію Fixed Balanced або перманентного режиму Survival, генеруючи ризик недооцінки реальної кіберзагрози.

4.4.3 Обмеження моделі батареї, оцінки Threat Level та апаратної платформи

Модель акумуляторної батареї, реалізовану в ASM-агенті, базують на концепції віртуального рівня заряду. Використано лінійну апроксимацію залежності швидкості розряду від поточного завантаження центрального процесора (CPU load), де емпіричні параметри DRAIN_IDLE та DRAIN_MAX визначають інтенсивність деградації заряду батареї за мінімального та максимального обчислювального навантаження відповідно. Такий підхід суттєво спрощує інтеграцію з апаратно-програмним стендом і створює можливість логічно відокремити алгоритми FLC від специфіки конкретної апаратної підсистеми електроживлення. Проте ця модель не враховує нелінійностей розряду реальних літій-іонних акумуляторів, температурних ефектів, впливу короткочасних пікових струмів та фізичного старіння елементів живлення, що є важливим фактором в енергообмежених IoT-вузлах. Для промислового впровадження розробленого методу необхідне апаратне калібрування цієї моделі під конкретну платформу (на основі прецизійних вимірювань струму та напруги) або пряма інтеграція фреймворку з наявною підсистемою моніторингу живлення для отримання фактичних апаратних значень заряду батареї.

Оцінку рівня кіберзагрози (Threat Level) у поточній реалізації формують переважно сценарно (через конфігураційні матриці) або на основі базових індикаторів мережевої активності (зокрема, інтенсивності вхідних HTTP-запитів, RPS). Такий підхід є цілком релевантним для експериментального HiL-моделювання DDoS-подібних аномалій, проте він не здатний замінити повноцінну інтеграцію з промисловими системами виявлення вторгнень (IDS) або ML-орієнтованими аналізаторами аномалій. Некоректна (хибно позитивна або хибно негативна) оцінка Threat Level на практиці здатна зумовити неадекватний вибір криптопрофілю: застосування недостатнього захисту під час реальної цілеспрямованої атаки або,

навпаки, невиправдану активацію ресурсомісткого профілю за безпечних умов, що порушить знайдений баланс між живучістю та безпекою.

Основні обмеження запропонованої моделі та потенційні шляхи їх усунення систематизовано в табл. 4.13.

Таблиця 4.13 – Основні обмеження запропонованої моделі

Характеристика обмеження	Сутність обмеження в межах поточної реалізації	Шляхи пом’якшення та перспективи вдосконалення
Емпірична модель батареї	Оцінка рівня заряду базується на лінійній залежності швидкості розряду від завантаження CPU. Не враховуються нелінійні характеристики акумуляторів, температурні ефекти та вплив старіння.	Апаратне калібрування моделі під конкретну платформу; пряма інтеграція з підсистемами моніторингу живлення для отримання апаратних значень рівня заряду батареї.
Оцінка рівня загрози (Threat Level)	Оцінка формується сценарно або на основі базових метрик інтенсивності (RPS). Існує ризик вибору неоптимального профілю через похибки ідентифікації складних кібератак.	Інтеграція адаптивного фреймворку з промисловими системами виявлення вторгнень (IDS) та інтелектуальними аналізаторами трафіку.
Апаратно-програмна специфіка	Калібрування виконано для ARM Cortex-A76 із застосуванням імітаційного синтетичного навантаження (багаторазове хешування) замість прямого виклику апаратно прискорених примітивів AES.	Проведення повторного профілювання навантаження (HiL-калібрування) під час перенесення на гетерогенні fog/edge-платформи без криптоприскорювачів.
Специфіка мережевого трафіку	Фреймворк і модель затримок орієнтовані на обробку великої кількості TCP/HTTP-трафіку з фіксованими характеристиками повідомлень.	Адаптація параметрів та проведення додаткових досліджень для протоколів CoAP/UDP, MQTT та промислових шин передачі даних.

Апаратно-програмна специфіка експерименту також накладає певні обмеження. Усі результати та калібрування енергетичних витрат отримано для платформи

Raspberry Pi 5 (архітектура ARM Cortex-A76). Оскільки сучасні процесори цього класу мають вбудовану апаратну підтримку криптографічних інструкцій (ARMv8 Cryptography Extensions), у дослідженні свідомо застосовано метод синтетичного обчислювального навантаження (багаторазове гешування сімейства SHA) для пропорційної імітації алгоритмів AES-GCM та ChaCha20-Poly1305. У разі перенесення ASM-фреймворку на інші мікроконтролерні або edge-платформи (особливо ті, що покладаються виключно на програмну реалізацію шифрування або використовують полегшені Ascon-подібні алгоритми), відносна енергетична вартість кожного криптопрофілю неминуче зміниться. Це вимагатиме повторного калібрування параметрів EnergyDelta. Крім того, поточну модель орієнтовано на оброблення TCP/HTTP-трафіку. Використання специфічних IoT-протоколів (CoAP/UDP, MQTT) потребуватиме додаткових досліджень щодо перерахунку часових затримок.

4.4.4 Стійкість до варіацій навантаження та шаблонів атак

Поведінку контролера керування безпекою (ASM-агента) в умовах зміни інтенсивності мережевого трафіку визначають через опосередкований вплив на вхідні змінні алгоритму – Battery Level та Threat Level. Збільшення інтенсивності запитів (RPS) прогнозовано підвищує середнє завантаження процесора. Згідно з реалізованою лінійною моделлю, це безпосередньо прискорює деградацію рівня заряду батареї. Водночас у сценаріях з імітацією кібератак зростання RPS або зміна сигнатур трафіку корелює з підвищенням показника Threat Level (режими нормального, підвищеного та DDoS-подібного навантаження).

Нечіткий логічний контролер (FLC) реагує на ці зміни комбіновано. В умовах високого рівня загрози та наявності достатнього енергоресурсу (сценарій Adaptive Full Battery) система активує обчислювально важкі профілі Balanced або Max. Однак за умови прогресивного розряджання батареї, навіть у разі збереження високого показника Threat Level, контролер безальтернативно переводить вузол до профілів

Survival або Balanced. Це свідомо знижує синтетичне криптографічне навантаження задля збереження живучості вузла.

Журнали реєстрації (часові діаграми) HiL-експериментів для різних шаблонів трафіку підтверджують стабільність закладеної логіки ухвалення рішень (рис. 4.5 та 4.7). Під час імітації масованої атаки (тривалий період високого Threat Level) система прогнозовано переходить до важчих криптопрофілів. Після нормалізації мережевого навантаження показник загрози знижується, і FLC автоматично повертає систему до енергоощадних режимів, що зменшує енергоспоживання. Завдяки застосуванню механізмів запобігання високочастотним перемиканням (anti-chattering), які реалізовано через обмеження частоти оновлення політики ASM-агентом та плавність функцій належності, активні профілі не осцилюють надмірно часто навіть за умов високої дисперсії вхідного навантаження.

У граничних (стресових) сценаріях експлуатації, коли Threat Level тривалий час залишається критично високим, а заряд батареї наближається до мінімально допустимої межі (сценарій Adaptive Low Battery, рис. 4.6 та 4.8), механізм примусово утримує систему в зонах Survival або Balanced. Це забезпечує базовий рівень конфіденційності й цілісності даних в умовах мінімально можливого навантаження на обчислювальне ядро. Зафіксована поведінка є практичним втіленням концепцій survivability-орієнтованої безпеки та керованої деградації для fog- та edge-інфраструктур. У таких умовах абсолютним пріоритетом стає підтримання операційної працездатності вузла, із чим запропонований адаптивний алгоритм успішно впорався під час тестування.

4.4.5 Практичні рекомендації щодо вибору ваг і політик для різних умов

Комплексний аналіз результатів, отриманих на базі експериментального ASM-стенду, дає змогу сформувати чіткі практичні рекомендації щодо фізичного налаштування нечіткого логічного контролера (FLC) та аналітичного вибору ваг інтегрального критерію якості J для різних класів застосувань fog- та IoT-вузлів (табл. 4.14).

Для енергообмежених мобільних вузлів (автономні дрони, переносні шлюзи, польові сенсори), де абсолютною вимогою є максимізація тривалості автономної роботи, необхідно налаштовувати параметри функцій належності на максимальне енергозбереження. Пороги для вхідної змінної Battery Level доцільно змістити вгору (наприклад, розширити зону LOW до 35%). Це гарантуватиме завчасну активацію профілю Survival, що забезпечить достатній резерв часу для безпечного завершення сесій і передавання критичних даних. Діапазон умов, за яких FLC дає змогу використовувати ресурсомісткий профіль Max, має бути максимально звуженим. Математичну валідацію та підтвердження ефективності такої алгоритмічної конфігурації слід здійснювати із застосуванням набору ваг Pro-Energy (з домінувальною вагою енергетичної складової w_P у цільовому критерії).

Для стаціонарних fog-вузлів із гарантованим електроживленням (промислові мережі, міська інфраструктура), де фундаментальним пріоритетом є безкомпромісний рівень захисту та цілісність сервісів, логіка контролера має бути дзеркально іншою. У таких сценаріях межі зони LOW мінімізують, або рівень заряду батареї взагалі аналізують лише як показник глибокого резерву. Пороги активації профілю Max у правилах FLC знижують для миттєвої реакції на підвищення Threat Level. Оцінювання якості функціонування такої системи здійснюють через Pro-Security ваговий набір (із домінуванням w_S), який математично підтверджує, що постійне використання важкої криптографії є виправданим компромісом.

Особливої уваги потребують змішані сценарії експлуатації (наприклад, edge-шлюзи, що живляться від електромережі, але мають резервні акумулятори на випадок відключень). Для таких вузлів найефективнішою є політика динамічної реконфігурації бази правил FLC. Під час стабільного електроживлення від мережі система застосовує безпеково-орієнтовані пороги (переважання профілів Max та Balanced). У разі фіксації переходу на резервний акумулятор агент автоматично зміщує пороги дискретизації, ініціюючи раннє включення режимів Survival/Balanced та жорстко обмежуючи використання Max. Для об'єктивного аналітичного супроводу та оцінювання ефективності такого гібридного переходу доцільно використовувати

Balanced-набір ваг ($w_L = 0.33, w_P = 0.33, w_S = 0.34$), який неупереджено фіксує компроміс між станами.

Рекомендовані набори ваг J та типові політики використання профілів для трьох класів сценаріїв представлені на табл 4.14.

Таблиця 4.14 – Рекомендовані набори ваг J та типові політики використання профілів Safe/Survival/Balanced/Max для трьох класів сценаріїв

Клас застосувань (Сценарій)	Налаштування алгоритму (пороги FLC для Battery Level)	Фізична політика використання криптопрофілів	Рекомендований набір ваг J (для валідації результатів)
Мобільні енергообмежені вузли (дрони, переносні шлюзи, польові сенсори)	Зміщені вгору (рання активація LOW). Наприклад:LOW < 35%,MEDIUM 35–70%,HIGH > 70%	Перевага надається профілю Survival для гарантування автономності. Діапазон активації Max максимально звужений для уникнення виснаження акумулятора.	Pro-Energy ($w_L = 0.1, w_P = 0.7, w_S = 0.7$)
Стаціонарні fog-вузли з гарантованим електроживленням (промислові мережі)	Знижені або ігноруються (акумулятор як крайній резерв). Наприклад:LOW < 10%,MEDIUM 10–30%,HIGH > 30%	Домінування профілю Max за високого рівня загрози (Threat Level = HIGH). Профілі Safe та Survival застосовуються вкрай рідко.	Pro-Security ($w_L = 0.1, w_P = 0.2, w_S = 0.7$)
Змішані edge-шлюзи зі змінним живленням (живлення від мережі / резервна батарея)	Динамічна зміна порогів залежно від активного джерела живлення. Базовий рівень:LOW < 20%,MEDIUM 20–60%,HIGH > 60%	Динамічна адаптація: під час живлення від мережі переважають Max/Balanced; під час переходу на батарею межі зміщуються для гарантованого переходу до Survival.	Balanced ($w_L = 0.2, w_P = 0.4, w_S = 0.4$)

Додатково варто зазначити, що у разі впровадження ASM-фреймворку на нових апаратних платформах (інша мікроархітектура ARM SoC, інша хімія акумуляторних

батарей або наявність апаратних криптоприскорювачів) обов'язковим етапом є коротке HiL-калібрування. Калібруванням є емпіричне вимірювання енергоспоживання для кожного застосованого криптопрофілю, уточнення констант DRAIN_IDLE та DRAIN_MAX у моделі батареї з подальшим коригуванням меж функцій належності.

Висновки до розділу

У четвертому розділі здійснено комплексну експериментальну перевірку та порівняльний аналіз розробленого методу енергоорієнтованого адаптивного керування криптографічним захистом. Для проведення досліджень синтезовано апаратно-програмний стенд апаратного моделювання (HiL, Hardware-in-the-Loop) на базі мікрокомп’ютера Raspberry Pi 5 (архітектура ARM Cortex-A76) із розгорнутою мікросервісною архітектурою фреймворку адаптивного менеджера безпеки (ASM). Застосування імітаційного синтетичного навантаження арифметико-логічного пристрою (АЛП) та програмної моделі розряджання акумуляторної батареї, яку параметризовано за фізичним завантаженням центрального процесора (CPU), забезпечило репрезентативне середовище для прецизійного вимірювання енергетичних, обчислювальних та QoS-метрик у єдиному експериментальному просторі.

Експериментально підтверджено обмеженість і неефективність застосування статичних політик безпеки (Fixed Max та Fixed Balanced) для автономних fog-вузлів. Встановлено, що відсутність реакції на динаміку енергетичних ресурсів та рівень кіберзагроз спричиняє нераціональне використання обчислювальних потужностей. Перманентне використання ресурсомістких криптопрофілів (Fixed Max) зумовлює критичне перевантаження процесора (до 53,69%) та пришвидшене виснаження заряду батареї (максимальний показник витраченої енергії EnergyDelta сягає 16,71%), тоді як фіксація на середньому рівні захисту залишає систему вразливою під час пікових атак (табл. 4.5, 4.6, 4.7).

Натомість результати досліджень довели високу апаратну ефективність запропонованого алгоритму керування на базі нечіткого логічного контролера (FLC). Найбільший практичний виграш від впровадження адаптивного механізму фіксують в умовах дефіциту енергетичного ресурсу (сценарій Adaptive Low Battery, рис. 4.6, 4.6). Підтверджено здатність системи до забезпечення режиму керованої деградації: у разі критичного падіння рівня заряду батареї контролер примусово знижує обчислювальне навантаження до профілів Survival або Balanced. Це стабілізує завантаження CPU на безпечному рівні (близько 31,06%) та дає змогу подовжити час автономної роботи fog-вузла під мережевим навантаженням в 1,7 раза порівняно зі статичною стратегією Fixed Max, не опускаючись нижче від базового мінімуму безпеки.

Комплексний аналіз експериментальних даних із застосуванням розробленого багатокритеріального інтегрального критерію якості J (табл. 4.10, 4.11) дозволив математично підтвердити адекватність та високу ефективність запропонованої системи адаптивного керування. Встановлено, що інтегральний критерій J виступає об’єктивним аналітичним інструментом оцінювання поточного стану Fog-вузла, інтегруючи в собі показники енергоспоживання, обчислювальної потужності та рівня захищеності.

Доведено, що за умов достатнього енергетичного резерву (високого рівня заряду акумуляторної батареї) алгоритм адаптивного керування забезпечує мінімізацію інтегрального показника J, упритул наближаючи параметри функціонування вузла до еталонних характеристик статичного профілю Max. Це свідчить про здатність системи підтримувати максимальний рівень криптографічної стійкості без загрози передчасного вичерпання енергетичного ресурсу.

Водночас у сценаріях критичного дефіциту енергії зафіксоване зростання розрахункового значення критерію J не є ознакою деградації алгоритму, а навпаки — відображає реалізацію свідомого та обґрунтованого інженерного компромісу. У таких режимах система адаптивного керування безальтернативно надає пріоритет фізичному виживанню апаратного вузла та збереженню його доступності в мережі над утриманням високих математичних оцінок стійкості захисту. Таким чином, зростання

показника J у складних умовах експлуатації є кількісним підтвердженням успішної адаптації системи, спрямованої на запобігання повному відключенню Fog-інфраструктури та забезпечення її відмовостійкості

За результатами досліджень формалізовано низку обмежень експериментальної методики (табл. 4.13), зокрема використання лінійної апроксимації розряджання батареї та прив'язку обчислювальної складності до архітектури ARM-ядра без застосування апаратних криптоприскорювачів. Це формує вимогу щодо обов'язкового апаратного HiL-калібрування струмів споживання в разі перенесення фреймворку на інші промислові платформи. На основі виявлених закономірностей сформовано комплексні практичні рекомендації (табл. 4.14) щодо налаштування меж функцій належності FLC та вибору вагових коефіцієнтів оцінювання. Запропонована методологія дає змогу інженерам ефективно адаптувати розроблений метод до різних класів IoT- та fog-систем – від енергообмежених автономних дронів до стаціонарних промислових шлюзів – без зміни фундаментального алгоритмічного ядра.

ВИСНОВКИ

У дисертаційній роботі розв’язано актуальне науково-прикладне завдання підвищення апаратної живучості та енергоефективності електронних вузлів туманних обчислень мобільного Інтернету Речей за умов автономного електроживлення шляхом розроблення та впровадження методу адаптивного керування режимами роботи мікропроцесора із застосуванням нечіткої логіки. Розроблена система дає змогу ефективно використовувати наявний обмежений енергетичний ресурс без потреби збільшення масо-габаритних показників пристрою.

Основні наукові та практичні результати дисертаційної роботи, отримані відповідно до поставлених завдань, полягають у такому:

1. Виконано системний аналіз режимів роботи Fog-вузлів як апаратних IoT-шлюзів щодо живучості за умов автономного електроживлення. З'ясовано, що виконання ресурсоємних обчислювальних алгоритмів (зокрема для забезпечення кібербезпеки) без застосування спеціалізованих апаратних прискорювачів потребує надмірних обчислювальних ресурсів ядра мікропроцесора. Це спричиняє стрімке зростання струмів споживання та швидке розряджання акумуляторної батареї. За результатами аналізу сформовано вимоги до енергоорієнтованого керування ресурсами, відповідно до яких головним пріоритетом має бути збереження фізичної працездатності електронного пристрою.

2. Сформульовано задачу адаптивного керування параметрами функціонування за змінних умов (поточного рівня заряду батареї та інтенсивності зовнішніх загроз). Визначено критерії оцінювання компромісу «безпека–затримка–енергія», а також введено й обґрунтовано використання комплексного показника ефективності (цільової функції), що дає змогу кількісно оцінити цей компроміс і є необхідним аналітичним інструментом для верифікації системи керування.

3. Розроблено метод адаптивного вибору режимів роботи мікропроцесора із застосуванням контролера нечіткої логіки. Цей метод практично реалізує інженерний принцип плавного зниження продуктивності: за критичних енергетичних режимів

система цілеспрямовано змінює обчислювальний профіль, унаслідок чого підтримує апаратну функціональність вузла та запобігає його раптовому знеструмленню.

4. Розроблено мікропрограмну архітектуру динамічного перемикання дискретних обчислювальних профілів для вбудованої платформи. Отримано контейнерну реалізацію архітектури, що узгоджує роботу криптографічних примітивів із відповідними керувальними рішеннями щодо рівня безпеки.

5. Спроєктовано та налаштовано програмно-апаратний стенд із застосуванням платформи з архітектурою ARM. Стенд інтегровано із зовнішнім генератором мережевого трафіку, що дало змогу забезпечити збір репрезентативних даних щодо струмів споживання за різних режимів роботи та верифікувати алгоритми керування.

6. Побудовано емпіричну модель енергоспоживання досліджуваної електронної системи, яка встановлює пряму залежність у ланцюгу «обчислювальна складність ↔ завантаження центрального процесора (CPU) ↔ струми споживання». Зазначену модель застосовано в методології апаратно-програмного моделювання, що створило об'єктивну базу для аналізу впливу режимів роботи ядра на інтенсивність розряджання батареї.

7. На апаратному стенді підтверджено ефективність розробленої системи керування: у штатному режимі («Adaptive Full Battery») витрата заряду становить 15,53 % порівняно з 16,71 % у статичному профілі, а у разі енергодефіциту (зниження заряду акумулятора до ≈ 35 %) алгоритм дає змогу знизити споживання до 10,04 % проти 16,71 %. Це зумовлює істотне підвищення живучості електронного вузла.

8. Окреслено межі застосування розробленої системи керування та визначено напрями подальшого розвитку експериментальної бази. Встановлено, що за відсутності виділених апаратних криптоприскорювачів перенесення розробленого фреймворку на інші промислові електронні платформи вимагає обов'язкового проведення процедури апаратного калібрування струмів споживання та налаштування меж функцій належності нечіткого контролера.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

[1] N. Siasi, M. A. Jasim, A. Yayimli, і N. Ghani, «Service Function Chain Survivability Provisioning in Fog Networks», *IEEE Transactions on Network and Service Management*, вип. 19, вип. 2, с. 1117–1128, Чер 2022, **DOI:** 10.1109/TNSM.2021.3138968.

[2] Y. Yilmaz, L. Aniello, і B. Halak, «ASSURE: A Hardware-Based Security Protocol for Resource-Constrained IoT Systems», *Journal of Hardware and Systems Security 2020 5:1*, вип. 5, вип. 1, с. 1–18, Вер 2020, **DOI:** 10.1007/s41635-020-00102-0.

[3] B. A. Mohammed *et al.*, «Taxonomy-Based Lightweight Cryptographic Frameworks for Secure Industrial IoT: A Survey», *IEEE Internet Things J.*, вип. 12, вип. 20, с. 43296–43316, 2025, **DOI:** 10.1109/JIOT.2025.3595649.

[4] S. M. Hashemi, A. Sahafi, A. M. Rahmani, і M. Bohlouli, «Energy-aware resource management in fog computing for IoT applications: A review, taxonomy, and future directions», *Softw. Pract. Exp.*, вип. 54, вип. 2, с. 109–148, Лют 2024, **DOI:** 10.1002/SPE.3273;JOURNAL:JOURNAL:1097024X;WGROUP:STRING:PUBLICATION.

[5] N. G. Zinabu, Y. W. Marye, K. K. Tune, і S. A. Demilew, «Comprehensive Analysis of Lightweight Cryptographic Algorithms for Battery-Limited Internet of Things Devices», *Int. J. Distrib. Sens. Netw.*, вип. 2025, вип. 1, с. 9639728, Січ 2025, **DOI:** 10.1155/DSN/9639728.

[6] B. Aslan, F. Yavuzer Aslan, і M. T. Sakalli, «Energy Consumption Analysis of Lightweight Cryptographic Algorithms That Can Be Used in the Security of Internet of Things Applications», *Security and Communication Networks*, вип. 2020, вип. 1, с. 8837671, Січ 2020, **DOI:** 10.1155/2020/8837671.

[7] J. Luc, A. O. Fobe, M. Nogueira, і D. Macêdo Batista, «A New Defensive Technique Against Sleep Deprivation Attacks Driven by Battery Usage», *Simpósio Brasileiro de Cibersegurança (SBSeg)*, с. 85–96, Вер 2022, **DOI:** 10.5753/SBSEG.2022.224911.

[8] A. Ghafara, P. Agung, A. Pradana, і R. Budiarto, «STATE OF THE ART ANALYSIS ON BATTERY-RELATED THREATS AND DEFENSES OF IOT DEVICES USING

KITCHENHAM», *Jurnal Teknik Informatika (Jutif)*, вип. 5, вип. 1, с. 179–189, Лют 2024, **DOI:** 10.52436/1.JUTIF.2024.5.1.1567.

[9] M. Barari і R. Saifan, «Energy–Aware security protocol for IoT devices», *Pervasive Mob. Comput.*, вип. 96, с. 101847, Груд 2023, **DOI:** 10.1016/J.PMCJ.2023.101847.

[10] H. V. Sampaio, F. Koch, C. B. Westphall, R. do N. Boing, і R. N. S. Cruz, «Autonomic Management of Power Consumption with IoT and Fog Computing», Трав 2021, **URL:** http://arxiv.org/abs/2105.03009.

[11] M. Saqib, B. Jasra, і A. H. Moon, «A Systematized Security and Communication Protocols Stack Review for Internet of Things», *2020 IEEE International Conference for Innovation in Technology, INOCON 2020*, Лис 2020, **DOI:** 10.1109/INOCON50539.2020.9298196.

[12] B. Palanisamy і J. Xu, «Efficient and Resilient Edge Computing: Algorithms, Techniques and Research Opportunities», *ACM International Conference Proceeding Series*, с. 8–11, Січ 2024, **DOI:** 10.1145/3631461.3632515;PAGE:STRING:ARTICLE/CHAPTER.

[13] M. R. Anawar, S. Wang, M. Azam Zia, A. K. Jadoon, U. Akram, і S. Raza, «Fog Computing: An Overview of Big IoT Data Analytics», *Wirel. Commun. Mob. Comput.*, вип. 2018, 2018, **DOI:** 10.1155/2018/7157192.

[14] W. Li, S. Cao, K. Hu, J. Cao, і R. Buyya, «Blockchain-Enhanced Fair Task Scheduling for Cloud-Fog-Edge Coordination Environments: Model and Algorithm», *Security and Communication Networks*, вип. 2021, 2021, **DOI:** 10.1155/2021/5563312.

[15] A. Liutkevičius, N. Morkevičius, A. Venčkauskas, і J. Toldinas, «Distributed Agent-Based Orchestrator Model for Fog Computing», *Sensors 2022, Vol. 22,* вип. 22, вип. 15, Сер 2022, **PMID:** 35957450.

[16] M. Medhat, K. El-Shafey, і A. Rashed, «IoT-fog based smart-building security system design and performance evaluation», *Journal of Computer Science*, вип. 16, вип. 9, с. 1325–1333, 2020, **DOI:** 10.3844/JCSSP.2020.1325.1333.

[17] R. Mahmud і A. N. Toosi, «Con-Pi: A Distributed Container-based Edge and Fog Computing Framework», *IEEE Internet Things J.*, вип. 9, вип. 6, с. 4125–4138, Сер 2021, **DOI:** 10.1109/JIOT.2021.3103053.

[18] M. A. Lawal, R. A. Shaikh, і S. R. Hassan, «An Anomaly Mitigation Framework for IoT Using Fog Computing», *Electronics 2020, Vol. 9,* вип. 9, вип. 10, с. 1–24, Вер 2020, **DOI:** 10.3390/ELECTRONICS9101565.

[19] F. Alsokhiry, A. Annuk, M. A. Mohamed, і M. Marinho, «An Innovative Cloud-Fog-Based Smart Grid Scheme for Efficient Resource Utilization», *Sensors 2023, Vol. 23,* вип. 23, вип. 4, Лют 2023, **PMID:** 36850350.

[20] R. Basir *et al.*, «Fog Computing Enabling Industrial Internet of Things: State-of-the-Art and Research Challenges», *Sensors 2019, Vol. 19,* вип. 19, вип. 21, Лис 2019, **PMID:** 31694254.

[21] F. Oliveira, P. Pereira, J. Dantas, J. Araujo, і P. MacIel, «Dependability Evaluation of a Smart Poultry House: Addressing Availability Issues Through the Edge, Fog, and Cloud Computing», *IEEE Trans. Industr. Inform.*, вип. 20, вип. 2, с. 1304–1312, Лют 2024, **DOI:** 10.1109/TII.2023.3275656.

[22] T. P. Dover, «Evaluating Medical IoT (MIoT) Device Security using NISTIR-8228 Expectations», Квіт 2021, **URL:** https://arxiv.org/pdf/2104.03283.

[23] N. Abbas, M. Asim, N. Tariq, T. Baker, і S. Abbas, «A Mechanism for Securing IoT-enabled Applications at the Fog Layer», *Journal of Sensor and Actuator Networks 2019, Vol. 8,* вип. 8, вип. 1, Лют 2019, **DOI:** 10.3390/JSAN8010016.

[24] I. Radhakrishnan, S. Jadon, і P. B. Honnavalli, «Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices», *Sensors 2024, Vol. 24,* вип. 24, вип. 12, Чер 2024, **PMID:** 38931791.

[25] M. Savka, «Analysis of the key models, methods, and means of data collection in the Internet of Things», *Technologies and Engineering*, вип. 26, вип. 2, с. 66–78, Лип 2025, **DOI:** 10.30857/2786-5371.2025.2.6.

[26] D. Kim, J. Yun, і D. Kim, «An Energy-Efficient Secure Forwarding Scheme for QoS Guarantee in Wireless Sensor Networks», *Electronics 2020, Vol. 9,* вип. 9, вип. 9, с. 1–15, Вер 2020, **DOI:** 10.3390/ELECTRONICS9091418.

[27] A. C. Djedouboum, A. A. A. Ari, A. M. Gueroui, A. Mohamadou, і Z. Aliouat, «Big Data Collection in Large-Scale Wireless Sensor Networks», *Sensors 2018, Vol. 18,* вип. 18, вип. 12, Груд 2018, **PMID:** 30567331.

[28] M. R. Mortada, A. Nasser, A. Mansour, і K. C. Yao, «In-Network Data Aggregation for Ad Hoc Clustered Cognitive Radio Wireless Sensor Network», *Sensors 2021, Vol. 21,* вип. 21, вип. 20, Жов 2021, **PMID:** 34695953.

[29] M. B. Apsara, P. Dayananda, і C. N. Sowmyarani, «A Review on Secure Group Key Management Schemes for Data Gathering in Wireless Sensor Networks», *Engineering, Technology and Applied Science Research*, вип. 10, вип. 1, с. 5108–5112, Лют 2020, **DOI:** 10.48084/ETASR.3213.

[30] O. Gurewitz, M. Shifrin, і E. Dvir, «Data Gathering Techniques in WSN: A Cross-Layer View», *Sensors 2022, Vol. 22,* вип. 22, вип. 7, Бер 2022, **PMID:** 35408263.

[31] I. S. M. Isa, M. O. I. Musa, T. E. H. El-Gorashi, і J. M. H. Elmirghani, «Energy efficient and resilient infrastructure for fog computing health monitoring applications», *International Conference on Transparent Optical Networks*, вип. 2019-July, Лип 2019, **DOI:** 10.1109/ICTON.2019.8840438.

[32] M. Kumar, M. Sethi, S. Rani, D. K. Sah, S. A. AlQahtani, і M. S. Al-Rakhami, «Secure Data Aggregation Based on End-to-End Homomorphic Encryption in IoT-Based Wireless Sensor Networks», *Sensors 2023, Vol. 23,* вип. 23, вип. 13, Лип 2023, **PMID:** 37448038.

[33] A. Salim, A. Ismail, W. Osamy, і A. M. Khedr, «Compressive sensing based secure data aggregation scheme for IoT based WSN applications», *PLoS One*, вип. 16, вип. 12, с. e0260634, Груд 2021, **PMID:** 34914706.

[34] M. K. Alasmari, S. S. Alwakeel, і Y. A. Alohali, «A Multi-Classifiers Based Algorithm for Energy Efficient Tasks Offloading in Fog Computing», *Sensors 2023, Vol. 23,* вип. 23, вип. 16, Сер 2023, **PMID:** 37631746.

[35] V. Jindal, «PERFECT STRATAGEM FOR QUERY PROCESSING IN WSN», *International Journal of Advanced Research in Computer Science*, вип. 9, вип. 2, с. 435–437, Квіт 2018, **DOI:** 10.26483/IJARCS.V9I2.5849.

[36] C. Tagliaro, M. Komsic, A. Continella, K. Borgolte, і M. Lindorfer, «Large-Scale Security Analysis of Real-World Backend Deployments Speaking IoT-Focused Protocols», *ACM International Conference Proceeding Series*, с. 561–578, Вер 2024, **DOI:** 10.1145/3678890.3678899.

[37] S. Al-Hamouz, «A Power Saving Hybrid Technique for IoT», *Mod. Appl. Sci.*, вип. 12, вип. 4, с. 30, Бер 2018, **DOI:** 10.5539/MAS.V12N4P30.

[38] M. Rosendo і J. Granjal, «Energy-Aware Security Adaptation for Low-Power IoT Applications», *Network 2022, Vol. 2, Pages 36-52*, вип. 2, вип. 1, с. 36–52, Січ 2022, **DOI:** 10.3390/NETWORK2010003.

[39] D. Silva, L. I. Carvalho, J. Soares, і R. C. Sofia, «A Performance Analysis of Internet of Things Networking Protocols: Evaluating MQTT, CoAP, OPC UA», *Applied Sciences 2021, Vol. 11,* вип. 11, вип. 11, Трав 2021, **DOI:** 10.3390/APP11114879.

[40] T. Saba, A. Rehman, K. Haseeb, S. A. Bahaj, і R. Damaševičius, «Sustainable Data-Driven Secured Optimization Using Dynamic Programming for Green Internet of Things», *Sensors 2022, Vol. 22,* вип. 22, вип. 20, Жов 2022, **PMID:** 36298227.

[41] S. Jeddou, F. Fernández, L. Diez, A. Baina, N. Abdallah, і R. Agüero, «Delay and Energy Consumption of MQTT over QUIC: An Empirical Characterization Using Commercial-Off-The-Shelf Devices», *Sensors 2022, Vol. 22,* вип. 22, вип. 10, Трав 2022, **DOI:** 10.3390/S22103694.

[42] R. Mustafa, N. I. Sarkar, M. Mohaghegh, і S. Pervez, «A Cross-Layer Secure and Energy-Efficient Framework for the Internet of Things: A Comprehensive Survey», *Sensors 2024, Vol. 24,* вип. 24, вип. 22, Лис 2024, **PMID:** 39598986.

[43] Y. A. A. S. Aldeen і K. N. Qureshi, «New trends in internet of things, applications, challenges, and solutions», *Telkomnika (Telecommunication Computing Electronics and Control)*, вип. 16, вип. 3, с. 1114–1119, Чер 2018, **DOI:** 10.12928/TELKOMNIKA.V16I3.8483.

[44] E. Otsetova-Dudin, «Handoff in Various Mobile Network Technologies», *Transport and Communications*, вип. 7, вип. 2, с. 28–32, Вер 2019, **DOI:** 10.26552/TAC.C.2019.2.6.

[45] S. Banerjee, A. K. Das, S. Chattopadhyay, S. S. Jamal, J. J. P. C. Rodrigues, і Y. Park, «Lightweight Failover Authentication Mechanism for IoT-Based Fog Computing Environment», *Electronics 2021, Vol. 10,* вип. 10, вип. 12, Чер 2021, **DOI:** 10.3390/ELECTRONICS10121417.

[46] A. Pradhan, S. Das, Md. J. Piran, і Z. Han, «A Survey on Security of Ultra/Hyper Reliable Low Latency Communication: Recent Advancements, Challenges, and Future Directions», Квіт 2024, **URL:** https://arxiv.org/pdf/2404.08160.

[47] М. Савка, «КОМПЛЕКСНА МОДЕЛЬ ЗБОРУ ДАНИХ В МЕРЕЖАХ МОБІЛЬНОГО ІНТЕРНЕТУ РЕЧЕЙ», *Science-based technologies*, вип. 66, вип. 2, с. 226–237, Лип 2025, **DOI:** 10.18372/2310-5461.66.20333.

[48] A. Ometov, O. L. Molua, M. Komarov, і J. Nurmi, «A Survey of Security in Cloud, Edge, and Fog Computing», *Sensors 2022, Vol. 22,* вип. 22, вип. 3, Січ 2022, **PMID:** 35161675.

[49] Z. Xie, X. Song, J. Cao, і S. Xu, «Energy efficiency task scheduling for battery level-aware mobile edge computing in heterogeneous networks», 2022, **DOI:** 10.4218/etrij.2021-0312.

[50] M. Avgeris, D. Spatharakis, D. Dechouniotis, A. Leivadeas, V. Karyotis, і S. Papavassiliou, «ENERDGE: Distributed Energy-Aware Resource Allocation at the Edge», *Sensors 2022, Vol. 22,* вип. 22, вип. 2, Січ 2022, **PMID:** 35062619.

[51] G. Patrizi *et al.*, «Electrical Characterization under Harsh Environment of DC-DC Converters Used in Diagnostic Systems», *IEEE Trans. Instrum. Meas.*, вип. 71, 2022, **DOI:** 10.1109/TIM.2021.3129513.

[52] J. Zidar, T. Matić, I. Aleksi, і Ž. Hocenski, «Dynamic Voltage and Frequency Scaling as a Method for Reducing Energy Consumption in Ultra-Low-Power Embedded Systems», *Electronics 2024, Vol. 13,* вип. 13, вип. 5, Лют 2024, **DOI:** 10.3390/electronics13050826.

[53] N. A. Angel, D. Ravindran, P. M. D. R. Vincent, K. Srinivasan, і Y. C. Hu, «Recent Advances in Evolving Computing Paradigms: Cloud, Edge, and Fog Technologies», *Sensors 2022, Vol. 22,* вип. 22, вип. 1, Груд 2021, **PMID:** 35009740.

[54] D. Kanellopoulos і V. K. Sharma, «Dynamic Load Balancing Techniques in the IoT: A Review», *Symmetry 2022, Vol. 14,* вип. 14, вип. 12, Груд 2022, **DOI:** 10.3390/sym14122554.

[55] B. Confais і B. Parrein, «Emerging Trends and Challenges in Fog and Edge Computing for the Internet of Things», *IoT 2022, Vol. 3, Pages 145-146*, вип. 3, вип. 1, с. 145–146, Лют 2022, **DOI:** 10.3390/iot3010009.

[56] K. G. Arachchige, P. Branch, і J. But, «Evaluation of Correlation between Temperature of IoT Microcontroller Devices and Blockchain Energy Consumption in Wireless Sensor Networks», *Sensors 2023, Vol. 23,* вип. 23, вип. 14, Лип 2023, **PMID:** 37514560.

[57] L. D. Hung і N. V. M. Thanh, «Implementation of ChaCha20-Poly1305 on Self-Organization Data Framing for Enhancing IoT Communication», *REV Journal on Electronics and Communications*, вип. 14, вип. 4, Груд 2024, **DOI:** 10.21553/rev-jec.368.

[58] R. Kalaria, A. S. M Kayes, W. Rahayu, E. Pardede, і A. Salehi Shahraki, «Adaptive context-aware access control for IoT environments leveraging fog computing», *Int. J. Inf. Secur.*, вип. 23, с. 3089–3107, 2024, **DOI:** 10.1007/s10207-024-00866-4.

[59] H. Xu, W. D. Liu, L. Li, D. J. Yao, і L. Ma, «FSRW: fuzzy logic-based whale optimization algorithm for trust-aware routing in IoT-based healthcare», *Scientific Reports 2024 14:1*, вип. 14, вип. 1, с. 16640-, Лип 2024, **PMID:** 39025873.

[60] U. Jain, S. Pirasteh, і M. Hussain, «Lightweight, secure, efficient, and dynamic scheme for mutual authentication of devices in Internet-of-Things-Fog environment», *Concurr. Comput.*, вип. 35, вип. 1, с. e7428, Січ 2023, **DOI:** 10.1002/CPE.7428;ISSUE:ISSUE:DOI.

[61] J. Singh, P. Singh, E. M. Amhoud, і M. Hedabou, «Energy-Efficient and Secure Load Balancing Technique for SDN-Enabled Fog Computing», *Sustainability 2022, Vol. 14,* вип. 14, вип. 19, Жов 2022, **DOI:** 10.3390/su141912951.

[62] S. Nachimuthu, F. Alsaif, G. Devarajan, і I. Vairavasundaram, «Real time SOC estimation for Li-ion batteries in Electric vehicles using UKBF with online parameter identification», *Scientific Reports 2025 15:1*, вип. 15, вип. 1, с. 1714-, Січ 2025, **PMID:** 39799173.

[63] T. Zhou, J. Shen, S. Ji, Y. Ren, і L. Yan, «Secure and Intelligent Energy Data Management Scheme for Smart IoT Devices», *Wirel. Commun. Mob. Comput.*, вип. 2020, вип. 1, с. 8842885, Січ 2020, **DOI:** 10.1155/2020/8842885.

[64] A. Giaretta, N. Dragoni, і F. Massacci, «IoT Security Configurability with Security-by-Contract», *Sensors 2019, Vol. 19,* вип. 19, вип. 19, Вер 2019, **PMID:** 31548501.

[65] S. Selvan і M. M. Singh, «Adaptive Contextual Risk-Based Model to Tackle Confidentiality-Based Attacks in Fog-IoT Paradigm», *Computers 2022, Vol. 11,* вип. 11, вип. 2, Січ 2022, **DOI:** 10.3390/computers11020016.

[66] A. Liutkevičius, N. Morkevičius, A. Venčkauskas, і J. Toldinas, «Distributed Agent-Based Orchestrator Model for Fog Computing», *Sensors 2022, Vol. 22,* вип. 22, вип. 15, Сер 2022, **PMID:** 35957450.

[67] T. A. Ahanger, U. Tariq, A. Ibrahim, I. Ullah, Y. Bouteraa, і F. Gebali, «Securing IoT-Empowered Fog Computing Systems: Machine Learning Perspective», *Mathematics 2022, Vol. 10,* вип. 10, вип. 8, Квіт 2022, **DOI:** 10.3390/math10081298.

[68] M. Tawfik, «Optimized intrusion detection in IoT and fog computing using ensemble learning and advanced feature selection», *PLoS One*, вип. 19, вип. 8, с. e0304082, Сер 2024, **PMID:** 39088558.

[69] S. Pavan *et al.*, «A Deep Intelligent Attack Detection Framework for Fog-Based IoT Systems», *Comput. Intell. Neurosci.*, вип. 2022, вип. 1, с. 6967938, Січ 2022, **DOI:** 10.1155/2022/6967938.

[70] M. Suárez-Albela, P. Fraga-Lamas, L. Castedo, і T. M. Fernández-Caramés, «Clock Frequency Impact on the Performance of High-Security Cryptographic Cipher Suites for Energy-Efficient Resource-Constrained IoT Devices», *Sensors 2019, Vol. 19,* вип. 19, вип. 1, Груд 2018, **PMID:** 30577524.

[71] P. Li, J. Su, і X. Wang, «ITLS/iDTLS: Lightweight end-to-end security protocol for iot through minimal latency», *SIGCOMM 2019 - Proceedings of the 2019 ACM SIGCOMM Conference Posters and Demos, Part of SIGCOMM 2019*, с. 166–168, Сер 2019, **DOI:** 10.1145/3342280.3342347.

[72] M. Suárez-Albela, T. M. Fernández-Caramés, P. Fraga-Lamas, і L. Castedo, «A Practical Evaluation of a High-Security Energy-Efficient Gateway for IoT Fog

Computing Applications», *Sensors (Basel)*, вип. 17, вип. 9, с. 1978, Вер 2017, **PMID:** 28850104.

[73] R. Kalaria, A. S. M. Kayes, W. Rahayu, E. Pardede, і A. Salehi Shahraki, «Adaptive context-aware access control for IoT environments leveraging fog computing», *International Journal of Information Security 2024 23:4*, вип. 23, вип. 4, с. 3089–3107, Лип 2024, **DOI:** 10.1007/s10207-024-00866-4.

[74] M. Suárez-Albela, P. Fraga-Lamas, і T. M. Fernández-Caramés, «A Practical Evaluation on RSA and ECC-Based Cipher Suites for IoT High-Security Energy-Efficient Fog and Mist Computing Devices», *Sensors 2018, Vol. 18,* вип. 18, вип. 11, Лис 2018, **PMID:** 30423831.

[75] W. Lardier, Q. Varo, і J. Yan, «Dynamic Reduced-Round Cryptography for Energy-Efficient Wireless Communication of Smart IoT Devices», *IEEE International Conference on Communications*, вип. 2020-June, Чер 2020, **DOI:** 10.1109/ICC40277.2020.9149305.

[76] S. Pallewatta, V. Kostakos, і R. Buyya, «MicroFog: A framework for scalable placement of microservices-based IoT applications in federated Fog environments», *Journal of Systems and Software*, вип. 209, с. 111910, Бер 2024, **DOI:** 10.1016/j.jss.2023.111910.

[77] S. Selvan і M. M. Singh, «Adaptive Contextual Risk-Based Model to Tackle Confidentiality-Based Attacks in Fog-IoT Paradigm», *Computers 2022, Vol. 11,* вип. 11, вип. 2, Січ 2022, **DOI:** 10.3390/computers11020016.

[78] P. Griffioen і B. Sinopoli, «Assessing Risks and Modeling Threats in the Internet of Things», Жов 2021, **URL:** http://arxiv.org/abs/2110.07771.

[79] S. P. K. Gudla, S. K. Bhoi, S. R. Nayak, і A. Verma, «DI-ADS: A Deep Intelligent Distributed Denial of Service Attack Detection Scheme for Fog-Based IoT Applications», *Math. Probl. Eng.*, вип. 2022, вип. 1, с. 3747302, Січ 2022, **DOI:** 10.1155/2022/3747302.

[80] S. Chu, J. Koe, D. Garlan, і E. Kang, «Integrating Graceful Degradation and Recovery through Requirement-driven Adaptation», *Proceedings - 2024 IEEE/ACM 19th*

Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2024, с. 122–132, Квіт 2024, **DOI:** 10.1145/3643915.3644090.

[81] A. Jevremovic, Z. Kostic, I. Chorbev, D. Perakovic, A. Shalaginov, і I. Cvitic, «Energy Efficiency of Kernel and User Space Level VPN Solutions in AIoT Networks», *IEEE Open Journal of the Computer Society*, вип. 6, с. 199–210, 2025, **DOI:** 10.1109/OJCS.2024.3522566.

[82] Y. V, «NeuroFuzzy-SecComm: A Neuro-Fuzzy Self-Evolving System for Secure Communication and Energy Optimization in Next-Gen 6G IoT Networks», с. 1–6, Січ 2026, **DOI:** 10.1109/iccca66364.2025.11325150.

[83] A. H. Zwiad, «A Lightweight Hybrid Framework for Secure Communication in Low-Resource IoT Devices», *Journal of Al-Qadisiyah for Computer Science and Mathematics*, вип. 17, вип. 4, с. 309-318–309–318, Груд 2025, **DOI:** 10.29304/jqcsm.2025.17.42564.

[84] A. Mehbodniya *et al.*, «Energy-Aware Routing Protocol with Fuzzy Logic in Industrial Internet of Things with Blockchain Technology», *Wirel. Commun. Mob. Comput.*, вип. 2022, вип. 1, с. 7665931, Січ 2022, **DOI:** 10.1155/2022/7665931.

[85] M. Hamzei, S. Khandagh, і N. Jafari Navimipour, «A Quality-of-Service-Aware Service Composition Method in the Internet of Things Using a Multi-Objective Fuzzy-Based Hybrid Algorithm», *Sensors 2023, Vol. 23,* вип. 23, вип. 16, Сер 2023, **PMID:** 37631769.

[86] I. Memon *et al.*, «Energy-Efficient Fuzzy Management System for Internet of Things Connected Vehicular Ad Hoc Networks», *Electronics 2021, Vol. 10,* вип. 10, вип. 9, Квіт 2021, **DOI:** 10.3390/electronics10091068.

[87] R. Zhang, V. E. Sathishkumar, і R. D. J. Samuel, «Fuzzy Efficient Energy Smart Home Management System for Renewable Energy Resources», *Sustainability 2020, Vol. 12,* вип. 12, вип. 8, Квіт 2020, **DOI:** 10.3390/SU12083115.

[88] J. Nowaková і M. Pokorný, «Intelligent Controller Design by the Artificial Intelligence Methods», *Sensors 2020, Vol. 20,* вип. 20, вип. 16, с. 1–27, Сер 2020, **PMID:** 32785005.

[89] M. S. Aslanpour, S. S. Gill, і A. N. Toosi, «Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research», *Internet of Things*, вип. 12, с. 100273, Груд 2020, **DOI:** 10.1016/j.iot.2020.100273.

[90] S. H. Rafat *et al.*, «Lightweight Cryptographic Algorithm Analysis for Secure IoT Communication on ESP-32 Platforms», *2025 IEEE International Conference on Quantum Photonics, Artificial Intelligence, and Networking, QPAIN 2025*, 2025, **DOI:** 10.1109/QPAIN66474.2025.11171742.

[91] A. N. Alvi, B. Ali, M. S. Saleh, M. Alkhathami, D. Alsadie, і B. Alghamdi, «Secure Computing for Fog-Enabled Industrial IoT», *Sensors 2024, Vol. 24,* вип. 24, вип. 7, Бер 2024, **PMID:** 38610310.

[92] N. Abbas, M. Asim, N. Tariq, T. Baker, і S. Abbas, «A Mechanism for Securing IoT-enabled Applications at the Fog Layer», *Journal of Sensor and Actuator Networks 2019, Vol. 8,* вип. 8, вип. 1, Лют 2019, **DOI:** 10.3390/jsan8010016.

[93] R. P. De Prado, S. García-Galán, J. E. Muñoz-Expósito, A. Marchewka, і N. Ruiz-Reyes, «Smart Containers Schedulers for Microservices Provision in Cloud-Fog-IoT Networks. Challenges and Opportunities», *Sensors 2020, Vol. 20,* вип. 20, вип. 6, Бер 2020, **PMID:** 32204390.

[94] C. Pillajo, C. Bustos, G. Salazar-Chacón, A. Estévez, і H. Pinto, «Energy Consumption Analysis in Networking: IPSec Tunnels Comparison with Different Encryption and Authentication Methods», *2nd International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE 2024*, 2024, **DOI:** 10.1109/ic-ETITE58242.2024.10493267.

[95] M. SAVKA, V. PILINSKY, «ADAPTIVE ENERGY-ORIENTED HARDWARE AND SOFTWARE SYSTEM FOR CRYPTOGRAPHIC PROTECTION OF INTERNET OF THINGS FOG NODES BASED ON FUZZY LOGIC», *Technologies and Engineering*, вип. 26(5), с. 73–90, Жов 2025, **DOI:** 10.30857/2786-5371.2025.5.6.

[96] N. Shalavi, A. Khoshsirat, M. Stellini, A. Zanella, і M. Rossi, «Accurate Calibration of Power Measurements from Internal Power Sensors on NVIDIA Jetson Devices»,

Proceedings - IEEE International Conference on Edge Computing, вип. 2023-July, с. 166–170, 2023, **DOI:** 10.1109/EDGE60047.2023.00034.

[97] D. Naseh, M. Abdollahpour, і D. Tarchi, «Real-World Implementation and Performance Analysis of Distributed Learning Frameworks for 6G IoT Applications», *Information 2024, Vol. 15, Page 190*, вип. 15, вип. 4, с. 190, Бер 2024, **DOI:** 10.3390/INFO15040190.

[98] M. Kasioulis, M. Symeonides, G. Pallis, і M. D. Dikaiakos, «Power Estimation Models for Edge Computing Devices», *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, вип. 14351 LNCS, с. 257–269, 2024, **DOI:** 10.1007/978-3-031-50684-0_20;WEBSITE:WEBSITE:DL-SITE;CTYPE:STRING:BOOK.

[99] M. J. Bocus, S. Qiu, R. J. Piechocki, і K. Eder, «Towards Automated and Predictive Network-Level Energy Profiling in Reconfigurable IoT Systems», *Proceedings of The 41st ACM/SIGAPP Symposium on Applied Computing (SAC'26)*, вип. 1, Жов 2025, **DOI:** XXXXXXX.XXXXXXX.

[100] H. Mohammed, F. Khalid, P. Sawyer, G. Cataloni, і S. R. Hasan, «InTrust-IoT: Intelligent Ecosystem based on Power Profiling of Trusted device(s) in IoT for Hardware Trojan Detection», *ACM International Conference Proceeding Series*, Жов 2021, **DOI:** 10.1145/3505253.3505262.

[101] S. Tofaily, I. Rais, і O. Anshus, «Representing Power Variability of an Idle IoT Edge Node in the Power State Model», *Proceedings - IEEE Congress on Cybermatics: 2023 IEEE International Conferences on Internet of Things, iThings 2023, IEEE Green Computing and Communications, GreenCom 2023, IEEE Cyber, Physical and Social Computing, CPSCom 2023 and IEEE Smart Data, Smart…*, с. 540–546, 2023, **DOI:** 10.1109/ITHINGS-GREENCOM-CPSCOM-SMARTDATA-CYBERMATICS60724.2023.00103.

Додаток А

Програмний код модуля адаптивного керування мікропроцесорним навантаженням на базі нечіткої логіки

```
import time
import psutil
from typing import Dict, Tuple
# Вагові коефіцієнти для інтегральної функції якості J
W_LATENCY = 0.2
W_BATTERY = 0.4
W_SECURITY = 0.4
class FuzzySecurityController:
    """Реалізація нечіткого логічного контролера типу Такаґі-Сугено нульового порядку
    (Takagi-Sugeno zero-order) згідно з розділом 3.2 дисертації."""
    def __init__(self):
        # Дискретні виходи для криптографічних профілів (формула 3.4)
        self.OUT_SAFE = 10.0
        self.OUT_SURVIVAL = 30.0
        self.OUT_BALANCED = 60.0
        self.OUT_MAX = 90.0
    def _trapmf(self, x: float, a: float, b: float, c: float, d: float) -> float:
        return max(min((x - a) / (b - a + 1e-9), 1.0, (d - x) / (d - c + 1e-9)), 0.0)
    def _trimf(self, x: float, a: float, b: float, c: float) -> float:
        return max(min((x - a) / (b - a + 1e-9), (c - x) / (c - b + 1e-9)), 0.0)
    def fuzzify_battery(self, bat: float) -> Dict[str, float]:
        """Фазифікація рівня заряду батареї (Battery Level)."""
        return {
            "LOW": self._trapmf(bat, -10, 0, 20, 40),
            "MED": self._trimf(bat, 20, 50, 80),
            "HIGH": self._trapmf(bat, 60, 80, 101, 110)
        }
    def fuzzify_threat(self, threat: float) -> Dict[str, float]:
        """Фазифікація індикатора рівня загрози (Threat Level)."""
        return {
            "LOW": self._trapmf(threat, -0.1, 0.0, 0.2, 0.4),
            "MED": self._trimf(threat, 0.3, 0.5, 0.7),
            "HIGH": self._trapmf(threat, 0.6, 0.8, 1.5, 2.0)
        }
    def compute(self, battery: float, threat: float) -> Tuple[str, float]:
        """Обчислення вихідного Security Score (дефазифікація за формулою 3.5)."""
        b_mem = self.fuzzify_battery(battery)
        t_mem = self.fuzzify_threat(threat)
        rules = [
            (min(b_mem["LOW"], t_mem["HIGH"]), self.OUT_SURVIVAL),
            (min(b_mem["LOW"], max(t_mem["LOW"], t_mem["MED"])), self.OUT_SURVIVAL),
            (min(b_mem["MED"], t_mem["LOW"]), self.OUT_BALANCED),
            (min(b_mem["MED"], t_mem["HIGH"]), self.OUT_BALANCED),
            (min(b_mem["MED"], t_mem["MED"]), self.OUT_BALANCED),
            (min(b_mem["HIGH"], t_mem["LOW"]), self.OUT_BALANCED),
            (min(b_mem["HIGH"], max(t_mem["MED"], t_mem["HIGH"])), self.OUT_MAX)
        ]
        numerator = sum(f_str * out_val for f_str, out_val in rules if f_str > 0)
        denominator = sum(f_str for f_str, _ in rules if f_str > 0)
        crisp_output = numerator / denominator if denominator != 0 else 0.0
        # Порогова дискретизація профілю (anti-chattering механізм)
        if crisp_output >= 80.0: profile = "max_security"
        elif crisp_output >= 50.0: profile = "balanced"
        elif crisp_output >= 20.0: profile = "survival"
        else: profile = "safe_shutdown"
```

```
        return profile, crisp_output
def compute_J(avg_latency_ms: float, battery: float, crisp_security: float) -> float:
    """
    Обчислення інтегральної функції якості J згідно з формулою (3.6).
    """
    L_norm = min(avg_latency_ms / 50.0, 1.0)
    P_norm = 1.0 - max(0.0, min(battery, 100.0)) / 100.0
    S_norm = crisp_security / 100.0
    return W_LATENCY * L_norm + W_BATTERY * P_norm + W_SECURITY * (1.0 - S_norm)
def update_battery_model(current_battery: float, cpu_load_percent: float) -> float:
    """Емпірична модель віртуального розряджання батареї згідно з формулою (3.1)."""
    DRAIN_IDLE = 0.02   # r_min
    DRAIN_MAX = 0.60    # r_max
    cpu_factor = cpu_load_percent / 100.0
    actual_degrade = DRAIN_IDLE + (DRAIN_MAX - DRAIN_IDLE) * cpu_factor
    return max(0.0, current_battery - actual_degrade)
```

Додаток Б

Програмна реалізація генератора обчислювального навантаження для апаратно-програмного стенда

```
import hashlib
import time

def simulate_cpu_load(profile: str, payload_size: int = 4096) -> None:
    """
    Імітаційне апаратне навантаження арифметико-логічного пристрою (АЛП)
    відповідно до складності криптографічних профілів (розділ 3.4.3).
    """
    buf = b"x" * payload_size

    if profile == "max_security":
        iters = 25000  # Імітація профілю Max (AES-256-GCM + HMAC-SHA-512)
        h = hashlib.sha512
    elif profile == "balanced":
        iters = 15000  # Імітація профілю Balanced (AES-128-GCM + HMAC-SHA-256)
        h = hashlib.sha256
    elif profile == "survival":
        iters = 6000   # Імітація профілю Survival (ChaCha20-Poly1305)
        h = hashlib.sha1
    else:
        iters = 1000   # Імітація профілю Safe Shutdown
        h = hashlib.md5

    digest = buf
    for _ in range(iters):
        digest = h(digest).digest()
```

Додаток В

Сценарій автоматизації апаратного експерименту

```
import time
import requests

# Конфігурація HiL-експерименту (згідно з методологією розділу 4.2.1)
EXPERIMENT_DURATION_SEC = 120
ATTACK_START_SEC = 40
ATTACK_END_SEC = 80

def execute_hil_scenario() -> None:
    """
    Автоматизація апаратного експерименту із застосуванням детермінованих
    фаз навантаження та релаксації.
    """
    start_time = time.time()

    while True:
        elapsed = time.time() - start_time
        if elapsed > EXPERIMENT_DURATION_SEC:
            break

        # Фазове керування загрозою (Threat Level)
        if ATTACK_START_SEC <= elapsed < ATTACK_END_SEC:
            set_external_threat_level(0.95)  # Фаза пікового навантаження (атака)
        else:
            set_external_threat_level(0.05)  # Фаза фонового режиму та релаксації

        generate_impulse_load()
        time.sleep(0.2)
```