КИЇВСЬКИЙ СТОЛИЧНИЙ УНІВЕРСИТЕТ ІМЕНІ БОРИСА ГРІНЧЕНКА

Кваліфікаційна наукова
праця на правах рукопису

ЧЕРНЕНКО РОМАН МИКОЛАЙОВИЧ

УДК 004.056.5

ДИСЕРТАЦІЯ

МОДЕЛІ ТА МЕТОДИ ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ ІНФОРМАЦІЇ, ЩО ПЕРЕДАЄТЬСЯ ВІДКРИТИМИ КАНАЛАМИ В МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ

Спеціальність 125 Кібербезпека та захист інформації

Подається на здобуття ступеня доктора філософії

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

Науковий керівник:
АНОСОВ Андрій Олександрович
кандидат військових наук, доцент

КИЇВ – 2024

АНОТАЦІЯ

Черненко Р.М. Моделі та методи забезпечення захисту інформації, що передається відкритими каналами в мережах інтернету речей. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 125 «Кібербезпека та захист інформації». – Київський столичний університет імені Бориса Грінченка, Київ, 2024.

Дисертація присвячена вирішенню актуального наукового завдання, сутність якого полягає в розробці методів криптографічного захисту інформації, що передається відкритими каналами зв'язку пристроями з обмеженими обчислювальними ресурсами, в мережах інтернету речей, для забезпечення підвищеної криптостійкості, імітостійкості та високої швидкості шифрування шляхом модифікації криптографічного алгоритму A5/1.

За даними звіту компанії Cisco, станом на 2023 рік, пристрої інтернету речей, вбудовані в звичне обладнання мікрокомп'ютери та контролери, які генерують та передають великі обсяги даних складають половину всіх пристроїв, що формують комп'ютерні мережі (близько 14,7 млрд).

Слід звернути особливу увагу на те, що значна кількість таких пристроїв використовується на об'єктах критичної інфраструктури для обробки та передачі інформації про стан об'єктів та ситуацію навколо них, про функціонування систем охорони та підтримки штатного функціонування. Зрозуміло, що в умовах повномасштабної війни такі мережі можуть бути використані державою-агресором в якості джерела розвідувальної інформації для досягнення власних терористичних цілей. Це актуалізує питання надійного захисту відповідних даних.

У той же час, виходячи з функціонального призначення таких мереж та виконуваних завдань їх складові, засоби та обладнання мають значно менші

обчислювальну потужність і ресурси, порівняно з поширюваними комп'ютерами та смартфонами.

Деякі з існуючих стандартних алгоритмів криптографічних перетворень (у тому числі – національні) хоча можуть бути реалізовані на частині відповідних апаратних та програмних платформ, але щодо цих реалізацій висуваються суттєві претензії щодо їх швидкодії. Зокрема, такі алгоритми не враховують особливості пристроїв, які мають менше 10 кБ оперативної пам’яті та менше 100 кБ вбудованої пам'яті (так звані пристрої класу С0).

Інше зауваження щодо відомих стандартів криптографічних алгоритмів (окрім національних) стосується прозорості їхнього проєктування, зокрема, недостатності інформації щодо умов їх безпечного застосування та управління криптографічними ключами.

Таким чином, існує актуальне не до кінця вивчене наукове завдання, яке полягає в площині вирішення проблеми безпечного шифрування даних на пристроях з обмеженими обчислювальними ресурсами.

Сутність цього завдання полягає в розробці моделей та методів захисту інформації, що передається відкритими каналами зв'язку пристроями з обмеженими обчислювальними ресурсами в мережах інтернету речей за допомогою криптографічних перетворень, що забезпечують підвищений рівень конфіденційності, криптостійкості, імітостійкості та високу швидкість шифрування на основі модифікації стандартного криптографічного алгоритму А5/1.

Для досягнення мети підвищення рівня безпеки інформації, що передається незахищеними каналами пристроями з обмеженими обчислювальними ресурсами в мережах інтернету речей, за рахунок розробки й впровадження моделей і методів криптографічного захисту інформації в таких мережах було отримано наукові результати:

1. Вперше запропоновано метод криптографічного захисту інформації в мережі інтернету речей на основі модифікованого алгоритму А5/1, що забезпечує підвищену стійкість шифрування та імітостійкість

завдяки застосуванню байтової обробки інформації та застосування вузла накладання шифру на основі змінного латинського квадрату. Алгоритм має високу швидкодію, яка на 30,70% більше ніж у відомих алгоритмів для інтернету речей.

2. Вдосконалено стандартний протокол Shockburst безпроводового інформаційного обміну в мережі, з метою безпечного формування сеансових ключів та забезпечення криптографічно захищеної передачі даних від пристроїв з обмеженими обчислювальними ресурсами до шлюзу.

3. Подальшого розвитку набула модель загроз для побудови системи захисту інформації, що обробляється пристроями з обмеженими обчислювальними ресурсами в мережі інтернету речей.

У вступі обґрунтовується актуальність та важливість теми дисертаційного дослідження, сформульована мета та задачі дослідження, визначено основні положення, а також наукову та практичну цінність отриманих результатів та зазначено особистий внесок автора.

У першому розділі проведено аналіз поточного стану дослідження наукової проблеми. Визначено необхідність та перспективи розробки методу криптографічного захисту інформації в мережі інтернету речей, що передається відкритими каналами зв'язку пристроями з обмеженими обчислювальними ресурсами. Сформульовані критерії для оцінки розроблюваного методу. Обґрунтовані мета та задачі дослідження.

У другому розділі були визначені критерії для аналізу існуючих алгоритмів криптографічних перетворень згідно спеціальних вимог NIST які здатні функціонувати на пристроях з обмеженими обчислювальними ресурсами. Було проведено дослідження криптоалгоритмів за визначеними критеріями. За результатами дослідження виявлено, що алгоритм Ascon який планується до стандартизації NIST є мало ефективним з точки зору продуктивності на 8- та 16-бітних пристроях класу C0. Найефективнішим алгоритмом виявився HIGHT, проте, що він забезпечує найбільшу пропускну здатність, він споживає більше всього оперативної пам’яті, та згідно наявних

досліджень має вразливості в безпеці. Побудовано модель загроз яка включає найімовірніші загрози стосовно інформації, що обробляється пристроями з обмеженими обчислювальними ресурсами.

У третьому розділі розроблено метод криптографічного захисту інформації за рахунок модифікації криптографічного алгоритму A5/1 для забезпечення комунікацій пристроїв інтернету речей. Модифікований алгоритм використовує байтову обробку інформації замість бітової та вузол накладання шифру на основі латинського квадрату (багатоалфавітна заміна). З використанням визначеного набору статистичних тестів показано, що шифруюча послідовність має рівномірний випадковий розподіл. Розроблено алгоритм реалізації методу та його окремих компонентів з урахуванням обчислювальних можливостей пристроїв класу C0 для забезпечення високої швидкості шифрування. Модифікований алгоритм має більшу довжину ключа та синхромаркер, який дозволяє відновити захищений зв'язок у випадку збою, без необхідності повторного надсилання ключа. Проведено аналіз поточного рівня інформаційного ризику, за яким визначено необхідність побудови криптографічного протоколу для ідентифікації пристроїв та управління ключами.

У четвертому розділі проведено дослідження платформ для реалізації модифікованого алгоритму та оцінки швидкодії. Обрано пристрої з обмеженими обчислювальними ресурсами, що за своїми характеристиками відносяться до пристроїв класу C0, які були визначені як ті, що потребують впровадження методу криптографічного захисту інформації. Вдосконалено стандартний протокол Shockburst безпроводового інформаційного обміну в мережі. Криптографічний протокол враховує надійну ідентифікацію пристроїв, безпечне управління ключами та скорочує кількість критичних криптографічних параметрів, що зберігаються на пристрої. Практично реалізовано систему інтернету речей з впровадженням модифікованого алгоритму. Проведено аналіз роботи та проведена оцінка ефективності модифікованого алгоритму.

Узагальнюючим результатом проведених досліджень є розроблений метод криптографічного захисту інформації, що передається відкритими каналами зв'язку пристроями з обмеженими обчислювальними ресурсами в мережі інтернету речей за рахунок модифікованого алгоритму A5/1. Даний метод забезпечує підвищену імітостійкість, дозволяє забезпечити надійний рівень криптографічного захисту з використанням мінімальної кількості обчислювальних ресурсів та надає механізми відновлення шифрованого зв`язку за рахунок використання синхромаркера. Модифікований алгоритм для реалізації методу не вимагає великої кількості ресурсів та може використовуватись на пристроях класу C0. Він є ефективнішим від відомих рішень по таким показникам: має на 30,70% вищу швидкість шифрування в порівнянні з відомими алгоритмами для інтернету речей. При цьому споживає на 3,62% менше енергії та забезпечує підвищену імітостійкість шифрування.

Дисертація виконувалась в Київському столичному університеті імені Бориса Грінченка.

Результати наукових досліджень були використані на кафедрі інформаційної та кібернетичної безпеки імені професора Володимира Бурячка Київського столичного університету імені Бориса Грінченка в рамках науково-дослідної роботи: «Методи та моделі забезпечення кібербезпеки інформаційних систем переробки інформації та функціональної безпеки програмно-технічних комплексів управління критичної інфраструктури» (№ 0122U200483, КСУБГ, м. Київ).

Також результати наукових досліджень прийняті до впровадження в діяльність ТОВ «2Д3Д» та в ТОВ «Технологічні ІТ рішення».

Ключові слова: кібербезпека, захист інформації, загроза безпеці, уразливість, інформаційно-комунікаційна система, інтернет речей, IoT, пристрої з обмеженими обчислювальними ресурсами, криптографічний захист, сторонній вплив, імітостійкість, шифрування, швидкодія шифрування, латинський квадрат, підстановка заміни, алгоритм, канал зв'язку.

ANNOTATION

Chernenko R.M. Models and Methods of Ensuring the Protection of Information Transmitted Through Open Channels in Internet of Things Networks. – Qualification of scientific work on the rights of a manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 125 “Cyber Security and Information Protection” – Borys Grinchenko Kyiv Metropolitan University, Kyiv, 2024.

The dissertation is devoted to solving the urgent scientific problem of developing methods for cryptographic protection of information transmitted over open communication channels by devices with limited computing resources in the Internet of Things networks, to ensure enhanced cryptographic resistance, imitation resistance, and high encryption speed by modifying the cryptographic algorithm A5/1.

According to a report by Cisco, as of 2023, Internet of Things devices embedded in conventional equipment, microcomputers, and controllers that generate and transmit large amounts of data make up half of all devices forming computer networks (about 14.7 billion).

Special attention should be paid to the fact that a significant number of such devices are used in critical infrastructure facilities for processing and transmitting information about the status of objects and the surrounding situation, the functioning of security systems, and the support of normal operation. It is evident that in conditions of full-scale war, such networks can be exploited by an aggressor state as a source of reconnaissance information to achieve its own terrorist goals. This highlights the question of reliable protection of relevant data.

At the same time, based on the functional purpose of such networks and the tasks performed, their components, facilities, and equipment have much less computing power and resources compared to widespread computers and smartphones.

Some of the existing standard algorithms of cryptographic transformations (including national ones) can be implemented on some of the corresponding hardware and software platforms, but there are significant claims about these implementations regarding their speed. In particular, such algorithms do not take into account the features of devices that have less than 10 kB of random-access memory and less than 100 kB of built-in memory (so-called class C0 devices).

Another observation regarding known cryptographic algorithm standards (besides national ones) concerns the transparency of their design, particularly the insufficient information regarding the conditions for their secure application and cryptographic key management.

Thus, there exists a scientifically underexplored task that lies in addressing the issue of secure data encryption on devices with limited computational resources.

The essence of this task involves the development of models and methods for protecting information transmitted over open communication channels by devices with limited computational resources in Internet of Things networks through cryptographic transformations that ensure increased levels of confidentiality, cryptographic resistance, imitation resistance, and high encryption speed based on the modification of the standard cryptographic algorithm A5/1.

To achieve the goal of increasing the security level of information transmitted over unprotected channels by devices with limited computing resources in the Internet of Things networks, due to the development and implementation of models and methods of cryptographic information protection in such networks, the following scientific results were obtained:

1. For the first time, a method of cryptographic information protection in the Internet of Things network based on the modified A5/1 algorithm has been proposed, providing increased encryption strength and imitation resistance through byte-level data processing and the application of a cipher overlay node based on a variable Latin square. The algorithm demonstrates high speed, which is 30.70% higher than that of known algorithms for the Internet of Things.

2. Improved the standard Shockburst protocol for wireless information exchange in a network, aiming to securely generate session keys and ensure cryptographically protected data transmission from devices with limited computational resources to the gateway.

3. The model of threats for building an information protection system that is processed by devices with limited computing resources in the Internet of Things network has been further developed.

The introduction substantiates the relevance and importance of the topic of the dissertation research, formulates the purpose and objectives of the study, identifies the main provisions, as well as the scientific and practical value of the results obtained, and indicates the author's contribution.

The first section analyses the current state of research on the scientific problem. The necessity and prospects of developing a method for cryptographic protection of information in the Internet of Things network transmitted over open communication channels by devices with limited computing resources are determined. The criteria for evaluating the developed method are formulated. The aim and objectives of the study are substantiated.

In the second section, criteria were defined for analyzing existing cryptographic transformation algorithms by the special requirements of NIST that can function on devices with limited computing resources. A study of cryptoalgorithms was carried out according to the defined criteria. The results of the study revealed that the Ascon algorithm, which is planned to be standardized by NIST, is not very effective in terms of performance on 8- and 16-bit C0 class devices. The most effective algorithm is found to be HIGHT, however, despite providing the highest throughput, it consumes the most RAM and is vulnerable according to existing research. A threat model is constructed, including the most probable threats regarding information processed by devices with limited computational resources.

In the third section, a method of cryptographic protection of information is developed by modifying the A5/1 cryptographic algorithm to ensure the communication of IoT devices. The modified algorithm uses byte-based information

processing instead of bit-based and a cipher overlay node based on the Latin square (multi-alphabet replacement). Using a defined set of statistical tests, it is demonstrated that the encryption sequence exhibits a uniform random distribution. An algorithm for implementing the method and its components is developed, considering the computing capabilities of C0 class devices to ensure high encryption speed. The modified algorithm features a longer key length and a synchronization marker that allows for the recovery of secure communication in case of failure without the need for retransmitting the key. An analysis of the current level of information risk is carried out, according to which the need to build a cryptographic protocol for device identification and key management is determined.

In the fourth section, research on platforms for implementing the modified algorithm and evaluating its performance is conducted. Devices with limited computational resources classified as C0-class devices, requiring the implementation of the information cryptographic protection method, are selected. The standard Shockburst protocol for wireless information exchange in the network was improved. The cryptographic protocol takes into account reliable device identification, and secure key management, and reduces the number of critical cryptographic parameters stored on the device. The Internet of Things system with the implementation of the modified algorithm is practically implemented. The work is analyzed and the effectiveness of the modified algorithm is evaluated.

The general result of the conducted research is the developed method of cryptographic protection of information transmitted over open communication channels by devices with limited computational resources in Internet of Things networks, achieved through modification of the A5/1 algorithm. This method ensures enhanced imitation resistance, allows for reliable cryptographic protection using minimal computational resources, and provides mechanisms for recovering encrypted communication through the use of a synchronization marker. The modified algorithm for implementing the method does not require a large amount of resources and can be used on C0-class devices. It is more efficient than known solutions in the following aspects: it has a 30.70% higher encryption speed compared

to known algorithms for the Internet of Things. Additionally, it consumes 3.62% less energy and ensures enhanced imitation resistance of encryption.

The dissertation was carried out at the Borys Grinchenko Kyiv Metropolitan University.

The results of scientific research were used at the Department of Information and Cyber Security named after Professor Volodymyr Buriachok of Borys Grinchenko Kyiv Metropolitan University within the framework of research work: “Methods and Models for Ensuring the Cyber Security of Information Systems for Information Processing and Functional Security of Software and Hardware Complexes for Critical Infrastructure Management” (No. 0122U200483, KSUBG, Kyiv).

Also, the results of scientific research have been accepted for implementation in the activities of 2D3D LLC and Technological IT Solutions LLC.

Keywords: cyber security, information protection, security threat, vulnerability, information and communication system, Internet of Things, IoT, devices with limited computing resources, cryptographic protection, third-party influence, imitation resistance, encryption, encryption speed, Latin square, substitution replacement, algorithm, communication channel.

Наукові статті, опубліковані у наукових виданнях, включених на дату опублікування до переліку наукових фахових видань України:

1. Черненко, Р. М., Рябчун, О. П., Ворохоб, М. В., Аносов, А. О., & Козачок, В. А. (2021). Підвищення рівня захищеності систем мережі інтернету речей за рахунок шифрування даних на пристроях з обмеженими обчислювальними ресурсами. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 3(11), 124–135. https://doi.org/10.28925/2663-4023.2021.11.124135

2. Корнієць, В., & Черненко, Р. (2023). Модифікація криптографічного алгоритму a5/1 для забезпечення комунікацій пристроїв IoT. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 4(20), 253–271. https://doi.org/10.28925/2663-4023.2023.20.253271

3. Черненко, Р. (2023). Оцінка продуктивності алгоритмів легкої криптографії на обмежених 8-бітних пристроях. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 1(21). https://doi.org/10.28925/2663-4023.2023.21.273285

4. Черненко, Р. (2023). Генерація псевдовипадкових послідовностей на мікроконтролерах з обмеженими обчислювальними ресурсами, джерела ентропії та тестування статистичних властивостей. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 2(22), 191–203. https://doi.org/10.28925/2663-4023.2023.22.191203

Наукова стаття, опублікована у періодичному науковому виданні, проіндексована у базі даних Scopus

1. Chernenko, R., Anosov, A., Kyrychok, R., Brzhevska, Z., & Spasiteleva, S. (2022). Encryption Method for Systems with Limited Computing Resources. CEUR Workshop Proceedings, 3288, pp. 142-148. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85143827975&partnerID=40&md5=fc5b960373acefb92e4755f0a571afb2

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ESB – Enhanced ShockBurst

IoT – Internet of Things ‘Інтернет речей’

GE – gate equivalent ‘еквівалент логічного вентиля’

GSM – Global System for Mobile Communications ‘глобальна система мобільного зв’язку’

NIST – National Institute of Standards and Technology ‘Національний інститут стандартів і технологій’

RFID – radio frequency identification ‘радіочастотна ідентифікація’

STS – statistical test suite ‘Набір статистичних тестів’

XOR – Exclusive or ‘побітове додавання за модулем 2’

АЦП – аналогово-цифровий перетворювач

КПЗВ – криптографічні перетворення згідно спеціальних вимог

ПЗП – постійний запам’ятовуючий пристрій

ПООР – пристрої з обмеженими обчислювальними ресурсами

М2М – машинно – машинний зв’язок

ОЗП – оперативний запам’ятовуючий пристрій

РЛЗЗ – регістр з лінійним зворотнім зв’язком

ВСТУП

Актуальність теми Інтернет речей (Internet of things, далі IoT) – це мережа фізичних пристроїв, автомобілів, побутових приладів та інших об'єктів, що вбудовані в електроніку, програмне забезпечення, датчики, виконавчі механізми та пристрої зв’язку, що дозволяє цим об'єктам взаємодіяти та обмінюватися даними. Кожен об'єкт ідентифікується за допомогою вбудованого контролера та здатний взаємодіяти в існуючій інфраструктурі інтернету. Технологія IoT є похідною від розподілених обчислень, штучного інтелекту, машинного навчання та інших передових технологій. Її також називають всеосяжною комп’ютерною мережею. У відповідності до звіту Cisco щодо розвитку мереж інтернету, станом на 2023 рік кількість пристроїв типу машина-машина (M2M) становитимуть половину глобальних підключених пристроїв і з’єднань, що становить близько 14,7 млрд пристроїв.

Об'єкти в IoT (пристрої, транспортні засоби, побутові прилади тощо) з'єднані один з одним за допомогою вбудованих мікросхем і датчиків, які передають великі обсяги даних. Ці пристрої також часто відносять до типу M2M. Всі дані з пристроїв збираються на шлюзі, а з цього шлюзу дані передаються до серверів.

Хоч це нова технологія, існує ще багато проблем, які потрібно вирішити, таких як безпека, конфіденційність, зв'язок, сумісність, довговічність, розробка, стандарти тощо. У IoT пристрої, датчики, шлюзи тощо дуже компактні, зазвичай споживають мало енергії та мають не великі обчислювальні ресурси, оскільки основними характеристиками IoT є швидкість, висока продуктивність та енергоефективність. Для захисту цих пристроїв потрібні надійні моделі та методи. Стандартні криптографічні алгоритми (AES, DES, RC5 тощо) не можуть бути ефективно реалізовані через те, що велика кількість IoT пристроїв мають обмежені обчислювальні ресурси, тобто мають значно менше об’єму оперативної пам’яті (ОЗП), постійної енергонезалежної пам’яті (ПЗП), мають нижчу розрядність та частоту процесора в порівнянні з класичними комп’ютерами, а також часто не мають

можливості підключення до мереж енергоживлення, відповідно системи з використанням таких пристроїв також мають обмеження.

Існують різні алгоритми криптографічних перетворень які можуть бути реалізовані на пристроях з обмеженими обчислювальними ресурсами (ПООР), проте не всі з них є ефективними з точки зору продуктивності зокрема швидкодії. Використання неефективних алгоритмів захисту може призвести до недостатнього рівня захисту інформаційних систем та порушенням їх роботи через брак необхідних ресурсів. Тому, розробка нових моделей та методів захисту інформації, що передаються відкритими каналами зв'язку ПООР в мережах IoT є важливою задачею для забезпечення безпеки інформаційної системи.

Тема «моделі та методи забезпечення захисту інформації, що передається відкритими каналами в мережах інтернету речей» є актуальною, оскільки технології IoT показують найбільший приріст серед всіх інших цифрових пристроїв, та є невід'ємною частиною повсякденного життя та бізнесу. Проте не можуть бути надійно захищені з використанням стандартних сучасних алгоритмів шифрування, в тому числі через низьку швидкість шифрування та часто недостатнього обсягу інформації щодо умов їх застосування. Порушення безпеки в таких системах може призвести до значних збитків, особливо в умовах їх застосування на об'єктах критичної інфраструктури.

Класичні моделі системи передачі інформації з криптографічним захистом по відкритому каналу на яку орієнтуються більшість алгоритмів шифрування не враховують багатьох етапів функціонування, такий як: ініціалізація обладнання в IoT, генерація ключів шифрування.

Отже, розробка та впровадження моделі системи передачі інформації з шифруванням, що передається відкритими каналами в мережах інтернету речей пристроями з обмеженими обчислювальними ресурсами, може стати ефективним рішенням для підвищення рівня безпеки систем IoT. Така модель повинна дозволити шифрувати дані на пристроях де, через вимоги до

обчислювальних ресурсів, не можуть ефективно бути реалізовані інші алгоритми шифрування, та таким чином забезпечити конфіденційність даних, що передаються. Реалізовані методи повинні дозволити якісно підвищити захист ПООР, особливо класу С0, які мають менше 10 кБ ОЗП, та менше 100 кБ ПЗП. Пропоновані алгоритми повинні забезпечити використання мінімуму обчислювальних ресурсів для реалізації та роботи.

Вирішенню питань безпеки інформації та криптографічного захисту присвячено роботи багатьох провідних вчених, серед яких: Ігор Миколайович Коваленко, Михайло Євгенійович Шелест, Бурячок Володимир Леонідович, Іван Дмитрович Горбенко, Сергій Олександрович Гнатюк, Антон Миколайович Олексійчук, Людмила Василівна Ковальчук, Роман Васильович Олійников.

Зокрема дослідженню питань захисту інформації в мережах інтернету речей присвячено роботи вчених, серед яких: Ярослав Романович Совин, Олексій Анатолійович Смірнов, Марія Юріївна Родінко, Julia Borghoff, Anne Canteaut, Gregor Leander, Christof Paar, Saurabh Singh, Pradip Kumar Sharma, Christoph Dobraunig, Maria Eichlseder, Florian Mendel, Martin Schläffer.

Більшість робіт, що пропонуються цими науковцями включають розробку та удосконалення алгоритмів шифрування. Однак не завжди враховують особливості пристроїв класу С0 з точки зору обчислювальних ресурсів для забезпечення високої швидкості шифрування. Алгоритми розроблені для 64-бітної архітектури, як наслідок вимагають додаткових операцій на 8-бітних контролерах, вимагають великої частки обчислювальних ресурсів відносно доступних на пристроях класу С0. Окрім того такі алгоритми, при шифруванні коротких повідомлень, вимагають більше циклів на байт, та збільшують розмір зашифрованого повідомлення до розміру ключа.

Таким чином, з приведеного аналізу публікацій можна зробити висновок, що пропоновані методи не ефективні на пристроях класу С0. Вони здатні забезпечити надійний захист при використанні великої частини обчислювальних ресурсів та є повільними на таких пристроях.

У зв’язку з цим, існує необхідність вирішення *актуального наукового завдання,* сутність якого полягає в розробці моделей та методів захисту інформації, що передається відкритими каналами зв'язку пристроями з обмеженими обчислювальними ресурсами в мережах інтернету речей за допомогою криптографічних перетворень, що забезпечують підвищений рівень конфіденційності, криптостійкості, імітостійкості та високу швидкість шифрування на основі модифікації стандартного криптографічного алгоритму A5/1.

Зв’язок роботи з науковими програмами, планами, темами. Напрям дисертаційного дослідження безпосередньо пов’язаний з реалізацією доктрини інформаційної безпеки України, Стратегії інформаційної безпеки та Стратегії кібербезпеки України. Дисертація виконана відповідно до планів наукової і науково-технічної діяльності Університету Грінченка в рамках науково-дослідної роботи: «Методи та моделі забезпечення кібербезпеки інформаційних систем переробки інформації та функціональної безпеки програмно-технічних комплексів управління критичної інфраструктури» (06.22 – 06.27рр.) (реєстраційний номер: 0122U200483).

Мета і завдання дослідження.

Забезпечення безпеки інформаційних ресурсів в мережах інтернету речей, включаючи їх конфіденційність і цілісність, за рахунок розробки моделей і методів криптографічного захисту інформації, що передається пристроями з обмеженими обчислювальними ресурсами.

Відповідно до поставленої мети, для вирішення наукового завдання в роботі сформульовані такі часткові завдання дослідження:

1. Розробити метод криптографічного захисту інформації в мережі IoT на основі модифікації алгоритму A5/1 для забезпечення підвищеної стійкості шифрування та імітостійкості.

2. Побудувати криптографічний протокол інформаційного обміну в мережі для забезпечення безпечного формування сеансових ключів та забезпечення криптографічно захищеної передачі даних від ПООР до шлюза.

3. Побудувати модель загроз для розробки системи захисту інформації, що обробляється пристроями з обмеженими обчислювальними ресурсами в мережі IoT.

4. Дослідити ефективність методу захисту інформації на пристроях з обмеженими обчислювальними ресурсами із застосуванням модифікованого алгоритму шифрування на пристроях класу С0.

Об'єкт дослідження – забезпечення кібербезпеки мереж інтернету речей, що побудовані на основі пристроїв з обмеженими обчислювальними ресурсами.

Предметом дослідження є моделі та методи криптографічного захисту інформації, що передається відритими каналами зв'язку на пристроях з обмеженими обчислювальними ресурсами в мережах інтернету речей.

Методи дослідження. системний аналіз, елементарно-теоретичний та структурно-генетичний аналіз під час аналізу поточного стану дослідження наукової проблеми, визначенню необхідності та перспектив розробки методів захисту даних на ПООР, формуванні критеріїв для оцінки розроблюваного методу;

елементарно-теоретичний та структурно-генетичний аналіз, індукція, абдукція, моделювання, системно-структурний підхід, теорія ймовірностей і математична статистика під час розробки моделі захисту інформації на ПООР, відборі алгоритмів шифрування, генерації послідовності ключів, розробки алгоритму функціонування моделі та її окремих компонентів, аналізі та оцінці основних функцій алгоритму, аналізі поточного рівня інформаційного ризику, за яким визначено основні джерела загроз;

структурно-генетичний аналіз, моделювання, експеримент під час аналізу платформ та алгоритмів для проведення експерименту, обранні ПООР, що за своїми характеристиками відносяться до пристроїв класу С0, які були визначені як ті, що потребують впровадження методів криптографічного захисту інформації, реалізації моделі IoT з впровадженням розробленої моделі

системи передачі інформації з шифруванням, що передається відкритими каналами зв'язку ПООР, проведенні оцінки ефективності розробленої моделі.

Наукова новизна одержаних результатів

1. Вперше запропоновано метод криптографічного захисту інформації в мережі IoT на основі модифікованого алгоритму A5/1, що забезпечує підвищену стійкість шифрування та імітостійкість завдяки застосування байтової обробки інформації та застосування вузла накладання шифру на основі змінного латинського квадрату. Алгоритм має високу швидкодію, яка на 30,70% більше ніж у відомих алгоритмів для IoT.

2. Вдосконалено стандартний протокол Shockburst безпроводового інформаційного обміну в мережі з метою безпечного формування сеансових ключів та забезпечення криптографічного захисту передачі даних від пристроїв з обмеженими обчислювальними ресурсами до шлюзу.

3. Подальшого розвитку набула модель загроз для побудови системи захисту інформації, що обробляється пристроями з обмеженими обчислювальними ресурсами в мережі інтернету речей.

Практичне значення одержаних результатів полягає в тому, що в дослідженні запропоновано метод криптографічного захисту інформації на основі модифікованого алгоритму A5/1, що передається відкритими каналами зв'язку пристроями з обмеженими обчислювальними ресурсами в мережі інтернету речей. Модифікований алгоритм не вимагає великої кількості обчислювальних ресурсів та може використовуватись на пристроях класу C0. Він є ефективнішим від існуючих рішень по таким показникам: має на 30,70% вищу швидкість шифрування в порівнянні з існуючими алгоритмами для IoT. При цьому споживає на 3,62% менше енергії та забезпечує підвищену імітостійкість шифрування.

Результати досліджень прийняті до впровадження в діяльність ТОВ «2Д3Д», та в ТОВ «Технологічні ІТ рішення».

Особистий внесок здобувача. Всі наукові результати, що виносяться на захист, одержано здобувачем самостійно.

У статті «Підвищення рівня захищеності систем мережі Інтернету речей за рахунок шифрування даних на пристроях з обмеженими обчислювальними ресурсами» опублікованій у співавторстві, внесок Черненка Р.М. полягає у дослідженні алгоритмів та розробці прототипу системи IoT.

У статті «Encryption Method for Systems with Limited Computing Resources» опублікованій у співавторстві, внесок Черненка Р.М. полягає у імплементації методу шифрування та формуванні криптографічних параметрів.

У статті «Модифікація криптографічного алгоритму a5/1 для забезпечення комунікацій пристроїв IoT» опублікованій у співавторстві, внесок Черненка Р.М. полягає у побудові моделі загроз, модифікації алгоритму та побудові криптографічного протоколу.

Апробація результатів дисертації.

Основні теоретичні та практичні результати були представлені та обговорені в науковій конференції:

Workshop on Cybersecurity Providing in Information and Telecommunication Systems, CPITS (13 October 2022)

Публікації. За результатами проведеного дисертаційного дослідження було опубліковано у 5 наукових публікаціях, а саме: 4 наукових виданнях (з них 2 у співавторстві), включених на дату опублікування до переліку наукових фахових видань України, 1 стаття (з них 1 у співавторстві) у періодичному науковому виданні, проіндексованому у базі даних Scopus.

Структура та обсяг дисертаційного дослідження.

Дисертація складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 121 найменування на 16 сторінках. Загальний обсяг роботи становить 156 сторінок серед яких 119 сторінок основного тексту, 30 рисунків, 7 таблиць.

РОЗДІЛ 1 ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ЗАСАДИ ЗАХИСТУ ІНФОРМАЦІЇ, ЩО ПЕРЕДАЄТЬСЯ ВІДКРИТИМИ КАНАЛАМИ ЗВ'ЯЗКУ В МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ

1.1. Аналіз розвитку пристроїв інтернету речей типу M2M

Сучасний економічний та технічний розвиток супроводжується активним розвитком інформаційних технологій. Зокрема, розвиваються комп'ютерні мережі до яких можна підключити практично будь-яку річ, від розетки до автомобіля, сукупність таких пристроїв визначається як IoT.

IoT – це мережа взаємопов'язаних обчислювальних пристроїв, механічних і цифрових машин, предметів, яким надаються унікальні ідентифікатори та можливість передавати дані по мережі без необхідності взаємодії людини з комп'ютером. Це стало можливим завдяки появі дешевих мікрокомп'ютерів і поширенню безпроводових мереж. Практично будь-який об'єкт, можна перетворити на частину IoT. Підключення всіх цих різних об'єктів і оснащення їх датчиками додає рівень цифрового інтелекту пристроям, дозволяючи їм передавати дані в режимі реального часу без участі людини. IoT робить навколишній світ розумнішим і більш чутливим, об'єднуючи цифровий і фізичний всесвіт.

Оскільки кількість таких взаємопов'язаних пристроїв продовжує зростати щодня, зростає і кількість загроз і вразливостей безпеки, що постають перед цими пристроями. Безпека є однією з найважливіших технологічних проблем у галузі IoT. Безпека має багато аспектів, але ключовим є безпека передачі даних і зберігання даних у системах та їх додатках. Існує велика кількість досліджень з цього питання з безліччю проблем і запропонованих рішень; однак більшість існуючих робіт не дає всебічного огляду питань безпеки та конфіденційності даних в IoT. У цьому контексті важливими є

довіра та безпека, щоб протистояти різноманітним атакам, загрозам, вразливостям та руйнівним наслідкам для суспільства і компаній [96].

У відповідності до звіту компанії Cisco щодо розвитку інтернету [109] кількість пристроїв типу M2M становитимуть половину глобальних підключених пристроїв і з'єднань станом на 2023 рік (рис. 1.1). Тобто їх кількість зросла з 33% у 2018 році до 50% станом на 2023 року, що у загальному становить близько 14,7 млрд підключень M2M. Абсолютна більшість пристроїв мережі IoT відносяться до типу M2M. Навіть ті пристрої які начебто взаємодіють з користувачем, роблять це через зовнішні шлюзи – різноманітні хмарні сервіси та додатки.

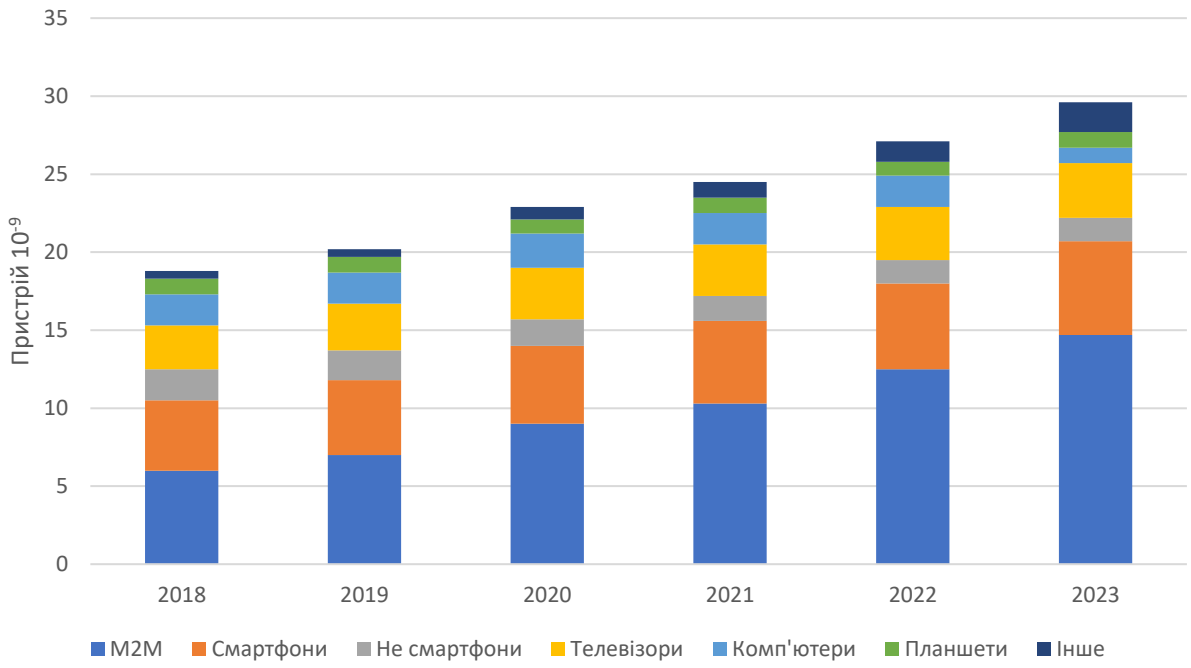

Рис. 1.1. Глобальний ріст кількості пристроїв до 2023 року

У відповідності до графіку M2M з'єднання розвиваються активніше за всі інші типи з'єднань і пристроїв. Ця категорія збільшиться в 2,4 рази протягом періоду та становить близько 50% всіх пристроїв на планеті. Таким чином забезпечення надійного захисту цих пристроїв є актуальною та пріоритетною задачею для дослідження.

M2M є досить широким визначенням, яке використовується для опису будь-якої технології, що дозволяє мережевим пристроям обмінюватися інформацією та виконувати дії без ручної допомоги людини. В таких підключеннях також може використовуватись штучний інтелект та машинне навчання, що в свою чергу дозволяє системі робити власні автономні вибори.

Технологія M2M вперше була прийнята в галузі машинобудування та промисловості, де інші технології, такі як диспетчерське управління і збір даних (SCADA) та віддалений моніторинг, допомагали віддалено керувати та контролювати дані отримані з обладнання. M2M з тих пір знайшло застосування в інших секторах, таких як охорона здоров'я, бізнес та страхування. Що найважливіше M2M є основою для IoT.

M2M можна визначити як автоматизовану передачу даних між пристроями. Іншими словами, після налаштування системи M2M-комунікація не повинна вимагати втручання людини. Наприклад, через M2M температурний датчик, розташований за сотні метрів під поверхнею океану, може збирати дані та передавати їх до центральної системи без необхідності втручання людини.

Завдяки M2M, комунікація вбудованих пристроїв є схожою на мобільну комунікацію. M2M є частиною IoT – в частині розуміння з’єднань між пристроями. Ethernet, мобільні мережі та інші технології публічних мереж допомагають IoT забезпечувати обмін інформацією M2M.

Одне із застосувань M2M – телеметрія датчиків, яка дозволяє компаніям віддалено зчитувати та контролювати різні показники певного середовища чи обладнання, як температура, енергоспоживання, вологість та тиск. За таким прикладом в Києві відбувається передача даних з мобільних систем контролю якості повітря. Для забезпечення конфіденційності, ідентифікації, цілісності, автентифікації, контролю доступу та імітостійкості переданих даних існує необхідність в шифруванні між датчиками в сучасних M2M мережах, а також у стандартних безпроводових сенсорних мережах.

Виникає ситуація коли система M2M зв'язку, що являє основу екосистеми IoT, може бути одним з найуразливіших компонентів цієї екосистеми [70]. Це обумовлено тим, що будь-які аномалії в поведінці мережі, ймовірно, залишаться непоміченими, оскільки зв'язок між з'єднаними машинами зазвичай відбувається без істотного людського нагляду. Крім того, скомпрометований M2M може поставити під загрозу фізичну інфраструктуру та безпеку через вплив системи на фізичне обладнання та контрольоване середовище [13].

Загалом IoT побудовано на базових ідеях M2M, розширивши їх до масштабних хмарних мереж пристроїв, які використовують хмарні мережні платформи для взаємодії один з одним. Вся інфраструктура, програмне забезпечення та платформа хмарної архітектури можуть використовуватися всіма пристроями IoT. Це дозволяє користувачам створювати мережі, які є швидкими, гнучкими та ефективними, об'єднуючи велику кількість пристроїв [80]. На рис. 1.2. зображено типову архітектуру IoT.

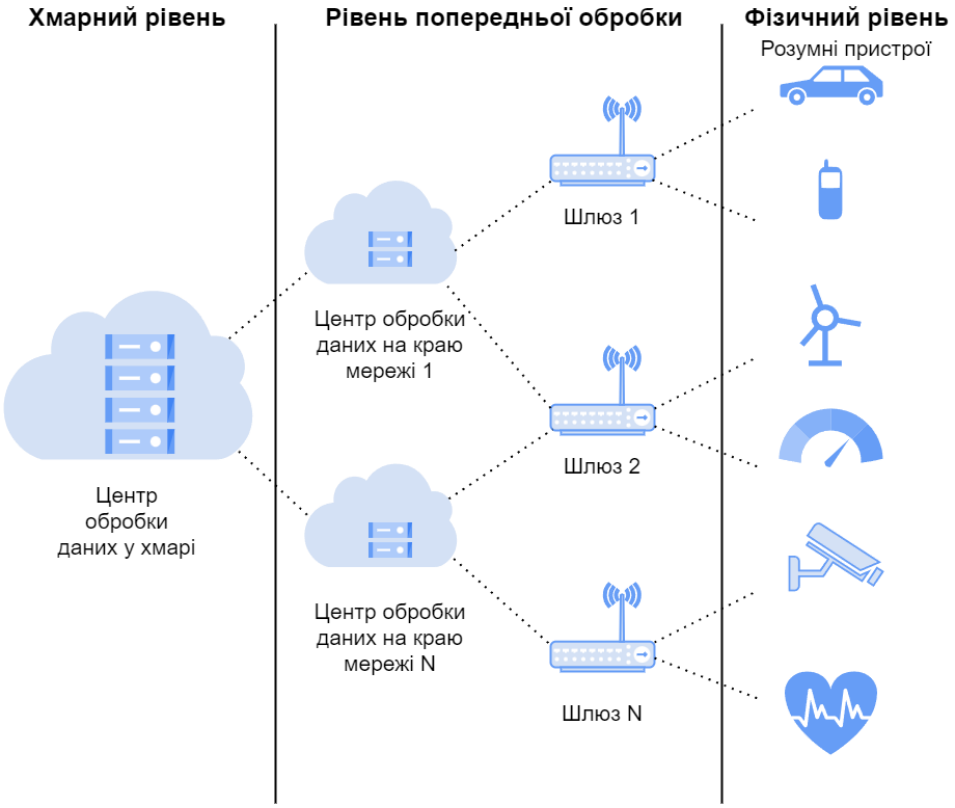

Рис. 1.2. Типова архітектура мережі інтернету речей

Іншим важливим критерієм є специфіка застосування таких пристроїв. У відповідності до прогнозу викладеним компанією Cisco, 48% таких пристроїв становитимуть побутові «розумні» пристрої, 25% пристрої для автоматизації підприємств, офісів та комерційних застосувань, 7% становитимуть пристрої медичного призначення, 6% інфраструктурні об'єкти міст, 6% пристрої розумних автомобілів, 3% видобуток енергії, 2% розумні енергетичні мережі та близько 3% інших пристроїв. При цьому найшвидше будуть розвиватись сфери пов'язані з автомобільним транспортом – 30% росту, автоматизації міст – 26% росту, енергетики 24% росту, медичного застосування – 19% росту.

Відповідно можна зробити висновок, що IoT все більше переходить зі сфери простої побутової автоматизації до індустрії 4.0 та знаходить своє застосування в процесах виробництва, контролю, налагодження, аналізу, критично важливих сферах функціонування компаній та навіть міст. Отже, питання захисту об'єктів, забезпечення конфіденційності, цілісності та доступності інформації є ключовими у функціонуванні критично важливих систем [31, 39].

Оскільки IoT є сферою яка стрімко розвивається, багато науковців працюють над забезпеченням кібербезпеки мереж IoT. В роботі [29] автори розглядають протоколи зв'язку як один з пріоритетів, в дослідженні [1] визначається пріоритетом ідентифікацію та локалізацію середовища. Але всі роботи зводяться до загального твердження – безпека повинна бути головним пріоритетом [4, 59].

Загалом дослідники виділяють три рівні IoT: фізичний рівень – тобто дії, що виконуються безпосередньо на пристроях та зазвичай стосуються збору даних та керування актуаторами, мережевий рівень – відповідає за підключення між пристроями IoT та іншими пристроями, та прикладний рівень – відповідає за надання послуг користувачу [77].

Авторами в [75] розглянуто найбільш розповсюджені вразливості пристроїв IoT, класифікованих за рівнями.

1. На **фізичному рівні** функціонують вузли збору даних. Цей рівень передбачає використання великої кількості різноманітних датчиків. Оскільки пристрої часто автономні, ці дані можуть бути скомпрометовані зловмисниками, або ж атака може бути направлена на фізичне функціонування пристроїв, якщо вони розміщені в незахищеному середовищі. Захист на цьому рівні може включати криптографічні методи, перевірку даних, зменшення кількості критичних параметрів, що зберігаються на пристрої, фізичний захист.

2. На **мережевому рівні** вразливості пов’язані з мережевими з’єднаннями пристроїв та передачею даних між ними. Типовими загрозами є нелегітимний доступ до мережі, перехоплення інформації, порушення конфіденційності, порушення цілісності, атаки типу людина по середині, атаки відмови в обслуговуванні. Окрім цього становище ускладнює, наявність великої кількості різних пристроїв від різних виробників, та використання мікрокомп’ютерів, що не мають достатньої кількості обчислювальних ресурсів для реалізації надійних криптографічних методів захисту, та те, що дані часто передаються відкритими каналами зв’язку. Також сюди можна віднести і частину фізичного рівня, з точки зору передачі даних між датчиками (які мають контролер) та іншими ПООР.

3. На **прикладному рівні** вразливості пов’язані з автентифікацією та правами доступу до даних. Це зумовлено великою кількістю виробників рішень для IoT, і їх програмною реалізацією різноманітних додатків. До цього рівня також можна додати і вразливості платформ для яких створюються ці додатки.

Можна зробити висновок, що вразливості на мережевому рівні часто є критичними, що зумовлено по-перше різноманітністю датчиків, мікроконтролерів та мікрокомп’ютерів, по-друге технічними характеристиками таких пристроїв, що не дозволяє реалізувати надійні методи захисту даних, або їх реалізація має суттєві недоліки, як низька швидкість, використання великої кількості обчислювальних ресурсів, по-третє

використання відкритих каналів зв’язку, що часто зумовлено специфікою використання таких пристроїв [97]. Окрім цього необхідність захисту систем типу M2M зумовлена через їх активний розвиток.

1.2. Стан дослідження проблеми захисту інформації, що передається відкритими каналами зв’язку в мережах інтернету речей

Канали зв'язку можна розглядати як засоби передачі інформації між пристроями та користувачами в мережі. Тобто сукупністю технічних засобів та фізичного середовища, що здатні передавати сигнали та забезпечують передачу даних від джерела інформації до отримувача.

За замовчуванням, канали зв'язку є незахищеними. Існують три основні критерії, що визначають безпечний канал зв'язку. Перший критерій – це переконання, що інформація, яка циркулює між користувачами, не повинна бути доступною для перегляду сторонніми особами. Другий критерій – це складність проникнення до каналу шляхом зламу паролів, використання шкідливого коду або використання вразливостей програмного інтерфейсу. Останній критерій полягає в тому, що канал зв'язку повинен бути безперебійним і постійно доступним, не маючи вразливостей, які можна використати.

M2M-пристрої для передачі інформації можуть використовувати різні комунікаційні технології в тому числі безпроводові [67], наприклад, IEEE 802.11 Wi-Fi, IEEE 802.15.4, LOWPANs або сервіс коротких повідомлень SMS, доступний за допомогою мобільних мереж стільникового зв’язку. Зокрема повідомлення з використанням SMS можуть містити команди управління, відомості про поточний стан системи, сигнал сповіщення, при цьому для SMS стандартні механізми захисту не передбачені [106].

Приватні та публічні безпроводові Wi-Fi-мережі, що розвинуті у всьому світі, а також мережі стандарту мобільного зв’язку ‘Global System for Mobile Communications’ (GSM), надають значний внесок у інтеграцію M2M-сенсорних пристроїв з інтернетом. Короткочасні комунікації між ПООР також можуть бути реалізовані за допомогою нових стандартів комунікації, таких як IEEE 802.15.4, особливо з урахуванням того, що багато M2M-застосувань можуть вимагати використання обмежених в частині живлення і можливостей обробки даних датчиків або актуаторів.

У контексті передачі інформації через відкриті канали зв'язку існує кілька основних проблем, які можуть виникнути і потенційно поставити під загрозу безпеку та конфіденційність інформації, що передається. Деякі з цих загроз включають:

1. Перехоплення: у відкритих каналах зв'язку інформація може бути піддана перехопленню третіми сторонами. Неавторизовані особи можуть прослуховувати та записувати передані дані, що загрожує конфіденційності та приватності.

2. Підробка: недостатній захист переданих даних, зокрема низький рівень імітостійкості, може відкрити можливість для підробки або зміни інформації під час передачі. Це може призвести до порушення цілісності даних і прийняття невірних рішень на основі зміненої інформації.

3. Доступність: відкриті канали зв'язку можуть бути вразливі до DoS атак, які можуть призвести до недоступності сервісу або обмеження доступу до переданої інформації.

4. Вразливості протоколів: використання криптографічно незахищених або застарілих протоколів відкритих каналів зв'язку може призвести до вразливостей, які можуть бути використані зловмисниками для отримання доступу до конфіденційної інформації або критичних параметрів.

5. Аутентифікація: відсутність або недостатня реалізація механізмів аутентифікації може призвести до можливості несанкціонованого доступу до переданої інформації.

Існує різноманіття підходів до захисту даних під час їх передачі через відкриті проводові та безпроводові канали зв'язку з метою зменшення ризику від порушення захищеності інформації. Значну роль у захисті даних, що можуть перебувати в різних станах, в тому числі під час передачі відіграють методи криптографічного захисту. Використання захищених протоколів, таких як SSL, TLS, FTP/FTPS, SSH, SSH протокол передавання файлів – SFTP, HTTPS та набір протоколів для захисту даних IP – IPSEC, захищають зміст даних під час їх передачі. Публічне ключове шифрування SSL/TLS забезпечує аутентифікацію та шифрування даних під час передачі.

Таким чином, якщо розглядати класичні мережі, то сучасні комп'ютери та інші електронні обчислювальні пристрої зазвичай мають достатньо обчислювальної потужності для реалізації стандартних, надійних алгоритмів шифрування та протоколів передачі даних, що знижує ризик несанкціонованого доступу до конфіденційної інформації.

Але стандартні алгоритми криптографічних перетворень здебільшого не можуть ефективно застосовуватись до мереж IoT, оскільки пристрої в таких мережах часто мають обмежені обчислювальні ресурси, до яких входять обмежена кількість ОЗП, низька частота та розрядність процесора, обмежена кількість ПЗП, обмеження пов'язані з енергоживленням [121]. Таким чином, вони не мають достатньо обчислювальної потужності для реалізації стандартних методів криптографічного захисту [38]. У RFC 7228 спеціальна група Internet Engineering Task Force визначила класи для ПООР [14], як показано в табл 1.1.

Таблиця 1.1

Класифікація пристроїв з обмеженими обчислювальними ресурсами

Клас	Розмір ОЗП, кБ	Розмір ПЗП, кБ
C0	< 10	< 100
C1	~ 10	~ 100
C2	~ 50	~ 250

Обчислювальні можливості таких пристроїв обмежують їх варіанти комунікації між собою або іншими пристроями. Для реалізації зв’язку між такими пристроями, надійне шифрування зазвичай не реалізується через їх обмежену потужність та обчислювальні ресурси, особливо пристроїв класу C0.

Розгортання невеликих обчислювальних пристроїв, таких як мітки радіочастотної ідентифікації ‘radio frequency identification’ (RFID), промислові контролери, сенсорні вузли та смарт-карти, стають все більш поширеними. Перехід від традиційних потужних комп'ютерів до невеликих малопотужних пристроїв призводить до широкого спектру нових проблем безпеки та конфіденційності. У багатьох конвенційних криптографічних стандартах, компроміс між безпекою, продуктивністю та вимогами до ресурсів був оптимізований для середовищ робочих станцій та серверів, і це ускладнює або робить неможливим їх реалізацію на ПООР. Хоча деякі стандартні алгоритми можуть бути реалізовані, їх продуктивність може бути неприйнятною для виконання функцій системи [73]. Це зумовлює необхідність пошуку нових методів захисту даних на ПООР.

Однією з передових ідей для забезпечення криптозахисту таких пристроїв є алгоритми криптографічних перетворень, що використовують прості операції з точки зору обчислень для підвищення швидкодії алгоритмів при застосуванні на ПООР. Зокрема спеціальні вимоги до таких алгоритмів були сформовані Національним інститутом стандартів і технологій ‘National Institute of Standards and Technology’ США (NIST). Криптографічні перетворення згідно спеціальних вимог (КПЗВ) NIST – це криптографічні алгоритми, методи або протоколи, адаптовані для реалізації в обмежених середовищах, включаючи RFID-мітки, датчики, безпроводові смарт-карти, медичні пристрої тощо [45].

Проте ідея таких полегшених, з точки зору обчислень, алгоритмів з’явилась з початком використання пристроїв, на яких не можуть бути

реалізовані стандарти криптографічного захисту, розроблені для традиційних комп’ютерних мереж.

Привід для розвитку КПЗВ NIST в основному випливає з їх прямого застосування в реальному світі, оскільки вони надають рішення для реальних проблем, з якими стикаються проєктувальники систем інтернету речей. Узагальнено, алгоритми КПЗВ NIST розроблені для досягнення двох основних цілей. Перша мета криптографічного алгоритму – протистояти всім відомим криптоаналітичним атакам і бути таким чином захищеним у моделі чорного ящика. Друга мета – побудувати криптографічний примітив таким чином, щоб його реалізації відповідали чітко визначеному набору обмежень, які залежать від конкретного випадку [40].

Зазвичай до методів і алгоритмів КПЗВ NIST ставляться такі вимоги [64]:

1. Алгоритми повинні працювати значно краще в обмежених середовищах (апаратні та вбудовані програмні платформи), порівняно із звичайними алгоритмами.
2. Вони повинні бути оптимізовані для ефективної роботи з короткими повідомленнями (наприклад, довжиною до 8 байт).
3. Повинні мати компактні апаратні реалізації та вбудовані програмні реалізації з низьким використанням ОЗП та ПЗП (в тому числі програмовані користувачем, з електричним стиранням).
4. Алгоритми повинні бути гнучкими, щоб підтримувати різні стратегії реалізації (низька енергія, низька потужність, низька затримка).
5. Продуктивність на мікроконтролерах повинна враховувати широкий спектр 8-бітних, 16-бітних і 32-бітних мікроконтролерних архітектур.
6. Попередня обробка та розгортання ключа (з точки зору часу обчислення та розміру пам'яті) повинна бути ефективною.

Окрім цього при розробці таких методів допускається пріоритизація продуктивності над іншими параметрами.

Це зумовлює необхідність розробки не вимогливих до обчислювальних ресурсів моделей та методів криптографічного захисту інформації, які могли б ефективно функціонувати на ПООР, особливо на пристроях класу С0 та забезпечувати підвищений рівень конфіденційності, криптостійкості, імітостійкості та високу швидкість шифрування.

1.3. Аналіз підходів до стандартизації передачі інформації засобів перевірки достовірності і управління пристроями передачі інформації

Як було визначено КПЗВ NIST повинні мати такі атрибути: потребують менше пам'яті, використовують невеликі обчислювальні ресурси та низьке енергоспоживання (у порівнянні зі стандартними криптографічними алгоритмами, такими як AES, DES, RSA), щоб забезпечити високу швидкодію та надійний захист, ПООР. Алгоритми КПЗВ NIST повинні бути простішими з точки зору обчислень та швидшими, ніж стандартні методи криптографічного захисту.

Стандартні криптографічні алгоритми перераховані вище хоч і є надійними [103], проте використовують великі ключі та складні алгоритми, які потребують багато енергії та обчислювальних ресурсів. IoT-пристрої часто працюють від акумуляторних елементів живлення і мають обмежені обчислювальні можливості, тому вони не можуть дозволити собі використовувати такі системи криптографії [2].

У 2012 році в дослідженні [13] було представлено шифр PRINCE. Він фокусується на апаратній реалізації. Шифр використовує 128-бітний ключ і складається з 64-бітного блоку з 12 раундів, S-блок цього шифру нелінійний. Принцип роботи шифру PRINCE полягає в тому, що він використовує структуру Файстеля для поділу блоку даних на два підблоки, які потім

шифруються окремо. Ключ шифрування використовується для шифрування кожного підблоку, після чого підблоки об'єднуються, щоб створити зашифрований блок. Шифр PRINCE має кілька переваг, включаючи низьку затримку, ефективну апаратну реалізацію та нелінійний S-блок. Однак він також має деякі недоліки, включаючи чутливість до атак, зв’язаних ключів, якщо використовується структура Файстеля з альтернативними ключами.

У [51] автори запропонували сімейство спрощених варіацій DES, які називаються DESL/DESX/DESXL. Основна ідея полягає в тому, щоб використовувати лише один S-блок рекурсивно, замість восьми різних S-блоків, щоб мінімізувати апаратну реалізацію.

DESL, DESX і DESXL мають ту ж структуру, що і DES, але вони використовують різні S-блоки. DESL використовує один S-блок, DESX використовує один S-блок, що генерується з восьми різних S-блоків DES, а DESXL використовує один S-блок, що генерується з восьми різних S-блоків DES і потім модифікується. Переваги нових варіацій DES включають: низьке споживання енергії, малу затримку і просту апаратну реалізацію. Проте мають і недоліки: зниження криптостійкості, оскільки вони використовують менше S-блоків.

У дослідженні [81] автори запропонували гібридний алгоритм LWC, який складається з двох алгоритмів LWC: швидкого і безпечного. Швидкий алгоритм використовується для шифрування і дешифрування невеликих даних, а безпечний алгоритм використовується для шифрування і дешифрування великих даних.

У статті [72] автори запропонували новий варіант XXTEA, який використовує покращений S-блок. Покращений S-блок робить XXTEA більш стійким до атак, на зв’язаних ключах, і атак з обраним текстом. Автори продемонстрували, що новий варіант XXTEA є більш стійким до атак, ніж оригінальний XXTEA. Також було продемонстровано, що новий варіант XXTEA є таким же швидким і ефективним, як оригінальний XXTEA. Новий варіант XXTEA може застосовуватись для ПООР, які потребують високої

безпеки. Проте варто зауважити, що він є досить повільним для шифрування великих об’ємів даних, або коли дані необхідно надсилати дуже швидко, особливо враховуючи пристрої класу C0.

У [97] у 2015 році було представлено Simeck. Simeck є блоковим шифром, який має 64-бітний блок і 12 раундів. Функція раунду Simeck складається з лінійних і нелінійних операцій. Проте він не такий надійний як деякі інші подібні, з точки зору споживання обчислювальних ресурсів, алгоритми шифрування.

Аналіз показав, що у цих дослідженнях не було розроблено унікального методу який би однаково добре працював на пристроях всіх класів. Як сказав – Керрі МакКей, спеціаліст по інформатиці NIST, – «Невеликі пристрої мають обмежені ресурси, і їм потрібна безпека, яка має компактну реалізацію. Ці алгоритми повинні охоплювати більшість пристроїв, які мають такі обмеження ресурсів» [63].

NIST було ініційовано у 2015 році проєкт з назвою Lightweight Cryptography ‘легковагова криптографія’ [54], метою якого є розробка алгоритмів шифрування які могли б працювати на більшості ПООР. В цьому проєкті були сформульовані КПЗВ NIST.

NIST почав досліджувати криптографію для обмежених середовищ у 2013 році. Після двох майстер-класів та обговорень зі стейкхолдерами з промисловості, урядом та академії, NIST започаткував процес конкурсу для оцінки та стандартизації схем, що забезпечують автентифіковане шифрування з пов'язаними даними та необов'язкові функції гешування для обмежених середовищ, де продуктивність поточних NIST-стандартів криптографії є неприйнятною.

NIST опублікував запит на пропозиції у 2015 році, і в 2016 році було отримано 17 пропозицій з алгоритмами шифрування для застосування на ПООР. Подані алгоритми були оцінені за кількома критеріями, включаючи безпеку, ефективність, простоту реалізації та можливість застосування в мережах з обмеженими обчислювальними ресурсами.

Проєктом були визначені такі критерії до розроблюваних методів криптографічного захисту [58]:

- повинні використовувати прості операції, наприклад, побітове додавання за модулем 2 ‘Exclusive or’ (XOR), поворот, перестановка біт, 4X4 S-блоки;
- повинні мати менші розміри блоків в порівнянні із традиційними алгоритмами шифрування інформації;
- повинні мати менші розміри ключів, але не менше 128 біт [64];
- повинні мати простіші схеми ключів, що в свою чергу призводить до простоти реалізації, проте і до пониження рівня захисту;
- можливість реалізації на апаратному рівні;

Під час розгляду потенційних моделей загроз були висунуті припущення. Методи криптографічного захисту для ПООР в тому числі КПЗВ NIST часто проєктуються з припущенням, що буде обмежена кількість відкритого тексту і шифротексту. Це пояснюється тим, що такі шифри часто використовуються на ПООР, які в свою чергу, мають обмежену тривалість роботи від батареї. Крім того, додаткового захисту надають криптографічні протоколи зв’язку.

Обґрунтування такого припущення виходить з наступного:

Обмеження пристроїв: пристрої з обмеженими обчислювальними ресурсами, такі як тривалість роботи від батареї, можуть не мати можливості зберігати велику кількість відкритого тексту і шифротексту.

Захист через протоколи: КПЗВ NIST часто використовуються в протоколах, які захищають від атак на зв’язаних ключах. Наприклад, протокол захисту на транспортному рівні, що використовує криптографічний нонс для запобігання атакам на зв’язаних ключах.

Припущення про обмежену кількість відомих даних і шифротексту дозволяє проєктувати шифри з меншими розмірами ключів і простішими алгоритмами. Це робить їх більш ефективними і простішими у впровадженні на ПООР.

Важливо зазначити, що це припущення не завжди є правильним. У деяких випадках може бути доступна велика кількість відкритого тексту і шифротексту. Це може зробити такі шифри уразливими до атак на зв’язаних ключах.

Тому важливо ретельно враховувати особливості функціонування мереж IoT при розробці алгоритмів КПЗВ NIST. Якщо є ризик наявності великого об’єму відкритого тексту і шифротексту, то слід використовувати більш надійні шифри.

Виходячи з таких вимог можна зробити висновок, що необхідно звести до мінімуму зберігання відкритого тексту та відповідного йому шифротексту та криптографічних параметрів на ПООР. В такому випадку можна використовувати простіші алгоритми. Оскільки для атак часто потрібна хоча б деяка кількість відкритих та зашифрованих даних.

Також були визначені суттєві загрози для ПООР:

– зловмисник потенційно може мати фізичний доступ до пристрою;
– такі пристрої часто є дешевими, відповідно проєктуються без надійних механізмів захисту проти атак по бічним каналам.

Враховуючи таке, існує потреба в розробці методів криптографічного захисту інформації, з врахуванням нейтралізації шкоди від фізичного порушення функціонування одного пристрою щоб це не призвело до загрози безпеки всій системі.

У березні 2017 року NIST опублікував NISTIR 8114, доповідь про КПЗВ NIST [58] і оголосив про створення портфелю «легковагових» алгоритмів шифрування через відкритий, конкурсний процес. У квітні 2017 року NIST опублікував чернетку профілю для стандартизації КПЗВ NIST [9] для збору відгуків щодо запропонованих функцій для первинного включення в портфель.

У травні 2018 року NIST опублікував повідомлення в Федеральному реєстрі, в якому посилався на проєкт вимог до подання та критеріїв оцінки для процесу стандартизації алгоритмів КПЗВ NIST, для громадського

обговорення. Вимоги та критерії оцінки були оновлені на основі громадських відгуків. У серпні 2018 року NIST опублікував повідомлення в Федеральному реєстрі, в якому оголошував про остаточні вимоги до подання та критерії оцінки для процесу стандартизації алгоритмів КПЗВ NIST та закликав до номінування криптографічних алгоритмів, які забезпечують автентифікацію з шифруванням з пов'язаними даними (AEAD) та необов'язкові функції гешування.

Процес подання алгоритмів КПЗВ NIST включав необов'язковий ранній процес рецензування, який мав на меті підвищити якість поданих матеріалів. До 25 лютого 2019 року, NIST отримав 57 пакетів заявок для розгляду на стандартизацію. З 57 заявок 56 були прийняті як кандидати першого раунду в квітні 2019 року, що ознаменувало початок першого раунду процесу стандартизації. Пакети заявок першого раунду були опубліковані на веб-сторінці проєкту NIST LWC для громадського огляду.

В ході першого раунду кількість кандидатів зменшилась з 56 до 32, щоб зосередити аналіз на більш перспективних алгоритмах. Кандидати другого раунду були оголошені 30 серпня 2019 року.

Оцінювання алгоритмів другого раунду ґрунтувалось на таких чинниках [87]:

Критерії оцінки: включали безпеку, ефективність, простоту реалізації, придатність для обмежених середовищ та прозорість.

Громадський відгук: NIST отримав громадський відгук на кандидатів першого раунду. Цей відгук був врахований при виборі кандидатів для другого раунду.

Внутрішній огляд: NIST провів внутрішній огляд кандидатів другого раунду. Цей огляд включав аналіз безпеки, ефективності та придатності для обмежених середовищ.

В результаті 2021 року NIST оголосив 10 фіналістів, які переходять до фінальної фази процесу відбору. Для фінального етапу були обрані такі

алгоритми шифрування: ASCON, Elephant, GIFT-COFB, Grain-128AEAD, ISAP, PHOTON-Beetle, Romulus, SPARKLE, TinyJAMBU та Xoodyak.

7 лютого 2023 року NIST оголосив про рішення стандартизувати сімейство ASCON для застосувань КПЗВ NIST. ASCON — це сімейство криптографічних алгоритмів, розроблених для використання на ПООР. Алгоритми ASCON були розроблені з урахуванням КПЗВ NIST, таких як простота реалізації, низьке енергоспоживання та стійкість до атак. Рішення було прийнято після ретельного аналізу різних пропозицій, які були отримані в рамках конкурсу. Було взято до уваги в тому числі громадський відгук на кандидатів, а також результати внутрішнього огляду.

NIST оголосив, що сімейство ASCON буде стандартизовано як стандарт КПЗВ NIST. Та буде використовуватися в різних застосунках, таких як IoT-пристрої, вбудовані системи та мобільні телефони.

Ascon — це алгоритм блочного шифрування з двома варіаціями: Ascon-128 і Ascon-128a. Він також є одним з алгоритмів у фінальному портфелі конкурсу CAESAR. Команда Ascon розробила нову варіацію, Ascon-80pq, яка захищена від квантового пошуку ключів.

Недоліком цього алгоритму є те, що він розроблений для використання на пристроях з 64-бітними процесорами, тоді коли існує величезна кількість пристроїв IoT з 8-, 12-, 16-, 32-бітними архітектурами, як наслідок при реалізації алгоритму на таких архітектурах збільшується кількість операцій до $n * \ell$, де n – розрядність машинного слова, ℓ - довжина тексту, за рахунок чередування бітів з їх сортуванням [23].

Також при шифруванні малих повідомлень, алгоритму необхідно набагато більше циклів на байт даних ніж для шифрування великих повідомлень [25]. Окрім цього, якщо повідомлення меншого розміру за ключ, то зашифрований текст буде дорівнювати саме довжині ключа, а не початкового повідомлення, що не є ефективним якщо система повинна надсилати багато коротких повідомлень.

Алгоритму потрібно близько 150 байт ОЗП для шифрування повідомлення розміром 16 байт, а також близько 10 кБ для реалізації алгоритму, що, наприклад, для платформ класу С0 є достатньо великим об’ємом. Так, як наприклад, контролери сімейства AVR ATmega 128 мають лише 2 кБ ОЗП та 32 кБ ПЗП [92].

Виходячи з такого, можна зробити висновок, що пропонований алгоритм шифрування може бути ефективним, з точки зору споживання обчислювальних ресурсів, варіантом для пристроїв класу С1 та С2. Особливо він є ефективним при розрядності процесора 64 біти, довжині машинного слова 64 біти, використанню рекомендованих параметрів та застосуванню 12 раундів шифрування. Проте враховуючи такі вимоги, до його ефективності на пристроях класу С0 можуть виникнути претензії щодо швидкодії, особливо у випадках надсилання великої кількості коротких повідомлень.

Розглянуті алгоритми є блоковими та орієнтуються на класичну модель системи передачі інформації з криптографічним захистом по відкритому каналу (рис. 1.3).

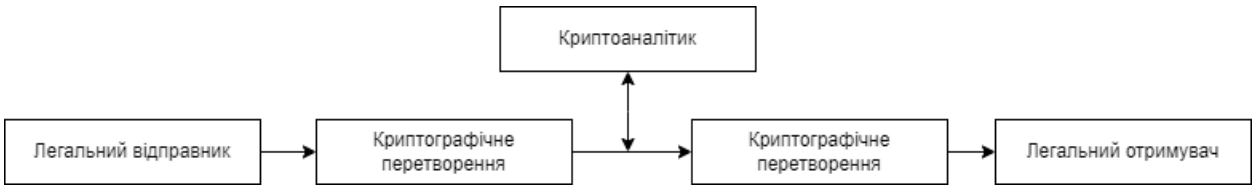

Рис. 1.3. Модель передачі інформації з криптографічним захистом

Однак, така модель не враховує багатьох етапів функціонування, таких як ініціалізація, генерація ключів шифрування, обмін ключами якщо необхідно. При цьому процеси генерації та керування ключами є критично важливими при реалізації потокових шифрів які за своєю природою можуть дозволити вирішити питання ефективного шифрування даних малого розміру на пристроях класу С0, за рахунок того, що не збільшують розмір повідомлення.

Окрім цього, автори пропонують тільки алгоритм шифрування та гешування, без загальної моделі функціонування системи на пристроях з обмеженими обчислювальними ресурсами. Зокрема недостатньо інформації, щодо управління криптографічними ключами та умов їх безпечного застосування.

Таким чином, це зумовлює необхідність розробки моделей та методів захисту інформації, що передається відкритими каналами зв'язку ПООР в мережі IoT за допомогою криптографічних перетворень, що забезпечують підвищений рівень конфіденційності, криптостійкості, імітостійкості та високу швидкість шифрування. При розробці таких методів необхідно враховувати обмеження пристроїв класу С0, щоб система могла функціонувати та забезпечувати надійний рівень захисту, при використанні мінімуму обчислювальних ресурсів, залишаючи їх для корисного навантаження. Повинна враховувати необхідність відправки коротких повідомлень. Розроблювані методи повинні мати чіткі вказівки, щодо формування секретних параметрів та схеми управління ключами.

1.4. Дослідження протоколів для передачі даних у мережі з низьким енергоспоживанням та обмеженими обчислювальними ресурсами

Зважаючи на те, що більшість пристроїв мережі IoT відносяться до типу M2M, є необхідність для подальшого дослідження, визначити які протоколи використовуються для передачі даних на ПООР в мережі IoT та M2M. Та які з них можуть бути реалізовані зокрема на пристроях класу С0.

У технологій M2M та мереж IoT немає стандартизованої технології комунікації між пристроями, і багато систем M2M призначені для виконання певних завдань або роботи з певними пристроями. Різні протоколи часто

використовуються в залежності від задач та доступних обчислювальних ресурсів, зокрема можна виділити наступні:

OMA DM (Open Mobile Alliance Device Management) – це протокол управління пристроями, який розроблений Open Mobile Alliance.

OMA LightweightM2M – це протокол управління пристроями, також розроблений Open Mobile Alliance. Він є більш легким варіантом протоколу OMA DM і призначений для використання з пристроями з обмеженими обчислювальними ресурсами, такими як датчики та актуатори.

MQTT – це протокол обміну повідомленнями, який використовується для передачі повідомлень між пристроями. Він є легким і ефективним протоколом і часто використовується в додатках M2M.

TR-069 (Technical Report 069) – це протокол прикладного рівня, який використовується для конфігурації та управління пристроями. Він широко використовується в IoT.

HyperCat – це протокол виявлення, який використовується для виявлення пристроїв та ресурсів у мережі. Він є ключовим компонентом архітектури OneM2M.

OneM2M – це архітектура зв’язку, яка розроблена для підтримки додатків M2M. Вона переважно, підтримується великою кількістю провайдерів мереж.

Google Thread – це протокол безпроводового інформаційного обміну, який розроблений Google. Він призначений для використання в додатках IoT і пропонує низький рівень енергоспоживання та легку конфігурацію.

AllJoyn – це відкрита платформа програмного забезпечення, яка розроблена для підтримки додатків IoT. Вона пропонує широкий спектр функцій, включаючи виявлення пристроїв, управління пристроями та обмін повідомленнями.

6LoWPAN (IPv6 поверх малопотужних безпроводових персональних мереж)[79] – стандарт та назва робочої групи IETF, що проєктує цей стандарт.

Основною ідеєю є забезпечення роботи безпроводових персональних мереж з мережами IP.

Якщо розглядати саме протоколи передачі даних, то в IoT часто використовуються такі протоколи:

– Wi-Fi – це протокол безпроводового інформаційного обміну, який використовується для передачі даних на коротких відстанях. Він широко використовується в додатках IoT, таких як розумні будинки та розумні міста;
– Bluetooth – це ще один протокол безпроводового інформаційного обміну, який використовується для передачі даних на коротких відстанях. Він використовується в додатках IoT, таких як розумні пристрої та пристрої які людина носить з собою;
– ZigBee – це протокол безпроводового інформаційного обміну низької потужності, який використовується для передачі даних на коротких відстанях. Він використовується в додатках IoT, таких як мережі датчиків і безпроводові мережі;
– LTE-M – це протокол безпроводового інформаційного обміну, який використовується для передачі даних на довгі відстані. Він розроблений для використання в додатках IoT, які вимагають широкого покриття;
– NB-IoT – це протокол безпроводового інформаційного обміну, який використовується для передачі даних на довгі відстані. Він розроблений для використання в додатках IoT, які вимагають низького енергоспоживання та широкого покриття;
– LoraWAN – це протокол передачі даних низької потужності, призначений для використання в додатках IoT. Він пропонує низьке енергоспоживання, довгий час роботи від батареї та широкий радіус дії.

Варто зауважити, що протоколи безпроводового інформаційного обміну часто вразливі до атак з використанням радіоперешкод, наприклад, Wi-Fi [85], Bluetooth [82], ZigBee [84].

Окрім цього, стандартні протоколи мереж, такі як HTTP, TCP і IP, можуть бути хорошим рішенням для додатків IoT, які мають такі характеристики:

– висока пропускна здатність: стандартні протоколи інтернету можуть передавати велику кількість даних за раз, що підходить для додатків, які працюють з великими обсягами даних [91];
– багатофункціональність: стандартні протоколи інтернету пропонують широкий спектр функцій, що підходить для додатків, які мають складні вимоги.

Проте варто зауважити, що реалізація стандартних протоколів можлива тільки у випадку достатньої кількості обчислювальних ресурсів, постійного джерела живлення, та наявності певної зони підключення.

Вибір протоколу для передачі даних в IoT залежить від різних факторів, таких як тип пристрою, відстань між пристроями, кількість даних, які потрібно передати, і енергоспоживання пристрою (рис. 1.4). У багатьох системах ПООР класу С0 спілкуються з мережею через шлюз, який діє як міст між ПООР та стандартною мережею [60].

Для ПООР на канальному рівні використовуються протоколи: Bluetooth, Thread, ZigBee Pro, ZigBee SE, NFC, LoraWAN.

ПООР здебільшого використовують протоколи 6LowPAN – UDP – DTLS – CoAP на верхніх рівнях моделі TCP/IP.

Високопродуктивні пристрої IoT здебільшого використовують IP v4/IP v6 – TCP/UDP – TLS – HTTP на вищому рівні та підтримують Wi-Fi, стільникові 2,5G/3G/4G (LTE) та Ethernet-технології.

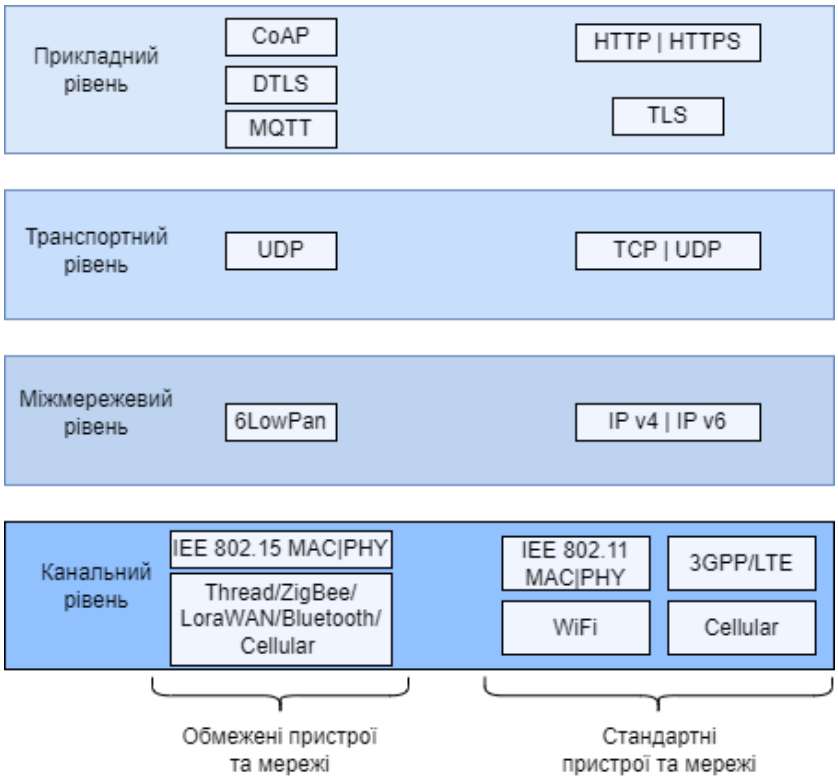

Рис 1.4. Відповідність протоколів потужності пристроїв

Один з протоколів який часто використовується ПООР класу C0 Enhanced ShockBurst (ESB) [7]. ESB є протоколом зв’язку, який підтримує двосторонню передачу даних у вигляді пакетів, має механізми буферизації пакетів, підтвердження та автоматичне передавання втрачених пакетів. ESB забезпечує радіозв'язок із низьким споживанням енергії, а його реалізація характеризується невеликим розміром пам’яті [65].

ESB підтримує підключення до восьми пристроїв передавачів до одного приймача за топологією «зірка». Транзакція за протоколом ESB є обміном пакетами між двома трансиверами, один з них є основним отримувачем, а інший використовується в ролі основного передавача. Пакети передані протоколом однозначно ідентифікуються за допомогою перевірки циклічним надлишковим кодом. Недоліком протоколу ESB є відсутність механізмів криптографічного захисту інформації при передачі.

Варто зауважити, що, наприклад, такі протоколи як Wi-Fi, 2,5G/3G/4G (LTE) та Ethernet-технології також можна реалізувати на ПООР, за умови

додаткових модифікацій пристроїв. Наприклад, впровадження додаткових модулів, які можуть бути потужнішими за самі ПООР, забезпечення достатнього джерела живлення, спрощення таких протоколів в тому числі в криптографічному плані для меншої кількості обчислювальних ресурсів для їх використання та реалізації.

В роботі [46] проаналізовані різні протоколи забезпечення зв’язку для IoT. Аналіз показав, що в даний час не існує єдиних критеріїв та методів вибору протоколів для інтеграції фізичних пристроїв від різних виробників з несумісними стеками протоколів та організації взаємодії між компонентами різних рівнів багаторівневої IoT-мережі. Аналіз вразливостей та можливих кібератак через протоколи зв'язку показав, що вони вразливі за рахунок неможливості застосування надійних алгоритмів шифрування зважаючи на обмеженість обчислювальних ресурсів приладів в ланці M2M, і це ставить під сумнів кібербезпеку всієї IoT-мережі, тому необхідно вжити додаткових заходів для підвищення захисту даних в таких мережах.

Таким чином, можна зробити висновок, що існує велика кількість протоколів, та відсутні стандарти щодо до їх використання. Багато протоколів, що забезпечують безпеку передачі даних не можуть бути реалізовані на ПООР, які в свою чергу складають велику частку мереж IoT, а ті що можуть не мають механізмів криптографічного захисту. Це зумовлює необхідність розробки методів криптографічного захисту інформації та побудови захищених протоколів передачі інформації, відкритими каналами зв'язку з використанням ПООР класу С0.

1.5. Методики аналізу інформаційних ризиків

Як було досліджено, значна частина IoT складається з ПООР, на яких через їх доступні обчислювальні ресурси неможливо ефективно реалізувати

стандартні методи, алгоритми та протоколи криптографічного захисту, окрім цього часто дані передаються саме відкритими каналами. Саме тому вони потребують розробки методів криптографічних алгоритмів та побудови протоколів, що адаптовані для реалізації в середовищах пристроїв на яких не можуть бути реалізовані стандарти безпеки розроблені для традиційних комп'ютерних мереж.

До таких методів, висуваються наступні вимоги, щоб реалізації відповідали чітко визначеному набору обмежень, які залежать від конкретного випадку:

Можливість виконання алгоритмів на ПООР. Такі методи та алгоритми повинні не просто мати змогу бути реалізованими на таких пристроях, а і забезпечувати високий рівень конфіденційності, криптостійкості, імітостійкості та високу швидкість шифрування. Звичайно, це залежить також від умов застосування, наприклад, від об'ємів та необхідної швидкості передачі інформації. Проте в будь-якому випадку алгоритми шифрування повинні залишати обчислювальні ресурси для виконання корисного коду, оскільки окрім збору даних мікроконтролери також можуть паралельно робити власні обчислення та керувати різними актуаторами та іншими системами. На практиці доведено, що перевантаження оперативної пам'яті на контролерах призводить до непередбачуваних результатів виконання коду. Таким чином існує необхідність використання простих з точки зору обчислень операцій для забезпечення криптографічних перетворень.

Іншою не менш важливою вимогою є енергоефективність [50]. Тобто методи повинні споживати дійсно мало енергії для виконання шифрування. Це зумовлено тим, що часто пристрої IoT живляться від вбудованих акумуляторів і їх робота повинна бути організована таким чином, щоб працювати якомога довше від одного заряду. Отож при розробці таких методів необхідно, щоб споживання енергії на шифрування певного об'єму було менше в порівнянні з іншими відомими методами, що забезпечують такий само рівень захисту або

більший. В такому випадку розроблюваний метод можна буде вважати ефективним.

Таким чином ці вимоги є основними в розробці такого методу, при цьому допустимо надавати пріоритет швидкодії та енергоефективності над рівнем захисту.

Для загальної оцінки ризиків можуть використовуватись стандартні затверджені методики оцінки ризиків такі як: ДСТУ ISO/IEC 27002:2023, NIST SP 800-30.

Методика NIST SP 800-30 є широко використовуваною в галузі інформаційної безпеки і є рекомендованою для використання багатьма організаціями. Вона є ефективною і зрозумілою у використанні, що робить її прийнятним вибором для організацій, які хочуть оцінити рівень ризику своїх ІБ-систем.

Переваги використання методики NIST SP 800-30:

- широко використовувана в галузі інформаційної безпеки;
- рекомендована багатьма організаціями;
- ефективна.

Але вона має також свої недоліки:

- не може забезпечити точну оцінку рівня ризику;
- не враховує всі фактори, що можуть вплинути на рівень ризику;
- може бути складною в використанні для складних систем.

Незважаючи на обмеження, методика NIST SP 800-30 є ефективним і практичним інструментом для оцінки рівня ризику ІБ-систем.

ДСТУ ISO/IEC 27002:2023 – це національний стандарт України, який встановлює вимоги до системи управління інформаційною безпекою. Стандарт базується на міжнародному стандарті ISO/IEC 27002:2023 і є його національним аналогом [108].

Система управління інформаційною безпекою – це система заходів, яка спрямована на запобігання, виявлення та реагування на загрози інформаційній безпеці. Вона складається з політики, процедур та процесів, які забезпечують

захист інформації від несанкціонованого доступу [30], використання, розголошення, модифікації або знищення.

Також можна послуговуватись «Методичними рекомендаціями щодо впровадження системи управління інформаційною безпекою та методики оцінки ризиків відповідно до стандартів Національного банку України» розробленими Національним банком України.

Окрім цього, необхідно проаналізувати компоненти криптографічного алгоритму, наприклад, шифруючу послідовність.

Криптографічні алгоритми вимагають випадкових чисел для генерації секретних ключів, одноразових кодів і ключів сеансу. Роль випадкової послідовності полягає в тому, щоб протистояти різним атакам і забезпечити безпеку даних. Криптографічні алгоритми поділяються на симетричні та асиметричні. Для генерації ключа в симетричному шифруванні потрібне випадкове число, тоді як асиметричне шифрування вимагає випадкових бітів для поєднання з вихідними даними. Геш-функції також потребують випадкових чисел для індексування та картування, щоб забезпечити цілісність та достовірність джерела даних.

Послідовність випадкових чисел для застосування в криптографії потребує чотирьох факторів, таких як відтворюваність, статистична незалежність, рівномірний розподіл і ефективність зберігання. Окрім цього, необхідно зважати на обмежені обчислювальні ресурси, тобто для пристроїв IoT генератор випадкових чисел повинен потребувати мінімуму обчислювальних ресурсів для реалізації. Генератор псевдовипадкових чисел надає відповідне рішення для генерації секретного ключа. Такі генератори схожі на криптографію, де користувач подає секретне «зерно» (початкове значення) у систему, і вона буде генерувати випадковий вихід (розпливчастий випадковий потік).

Для перевірки ймовірнісно-статистичних якостей шифруючої послідовності можуть використовуватись різні статистичні тести [105]. Тести зазвичай групуються в набори, щоб надати більш всебічне дослідження

випадковості. Існує три загальноприйняті набори тестів для аналізу випадковості: Набір статистичних тестів ‘statistical test suite’ (STS) від NIST [76], Dieharder[15] і TestU01[52].

NIST STS має особливе значення, оскільки він був опублікований як стандарт NIST і використовується для офіційної сертифікації. Тобто його можна вважати певним стандартом. Окрім цього документація NIST STS дає чіткі вказівки щодо інтерпретації результатів тестів, але все ж таки інтерпретація інколи використовує лише наближені значення.

Таким чином, розроблюваний метод повинен бути енергоефективним, ефективним з точки зору споживання обчислювальних ресурсів та забезпечувати підвищений рівень конфіденційності, криптостійкості, імітостійкості та високу швидкість шифрування інформації, що передається відкритими каналами зв’язку ПООР в мережах IoT.

Відповідно для аналізу рівня інформаційного ризику системи, можна використовувати такі загально прийняті стандарти як ДСТУ ISO/IEC 27002:2023. та NIST SP 800-30.

1.6. Обґрунтування мети та задач дослідження

Зважаючи на те, що ПООР використовуються в найрізноманітніших сферах, в агросекторі, в побуті, на об’єктах критичної інфраструктури, для обробки та передачі інформації про стан об’єктів та ситуацію навколо них, про функціонування систем охорони та підтримки штатного функціонування. Зрозуміло, що в умовах повномасштабної війни такі мережі можуть бути використані державою-агресором в якості джерела розвідувальної інформації для досягнення власних терористичних цілей. Це актуалізує питання надійного захисту інформації в таких мережах.

Проаналізувавши сучасні дослідження в цій галузі можна зробити висновок, що існуючі стандартні алгоритми криптографічних перетворень хоча і можуть бути реалізовані на частині ПООР класу С0, проте щодо таких реалізацій висуваються суттєві претензії щодо їх швидкодії. Існуючі алгоритми часто працюють ефективно на пристроях класу С1 та С2, при цьому є малоефективними з точки зору швидкодії та споживання ресурсів на пристроях класу С0, які становлять велику частку IoT за рахунок дешевизни та доступності. Такий стан обумовлено тим, що більшість криптографічних алгоритмів проєктуються без урахування обмежень в плані обчислювальних ресурсів пристроїв класу С0. Окрім цього, є зауваження щодо відомих стандартів криптографічних алгоритмів (окрім національних), що стосується прозорості їхнього проєктування, зокрема, недостатності інформації щодо умов їх безпечного застосування та управління криптографічними ключами.

У зв’язку з цим, існує необхідність вирішення актуального наукового завдання, сутність якого полягає в розробці моделей та методів захисту інформації, що передається відкритими каналами зв'язку ПООР в мережах IoT за допомогою криптографічних перетворень, що забезпечують підвищений рівень конфіденційності, криптостійкості, імітостійкості та високу швидкість шифрування на основі модифікації стандартного криптографічного алгоритму A5/1.

Метою дисертаційного дослідження є забезпечення безпеки інформаційних ресурсів в мережах IoT, включаючи їх конфіденційність і цілісність, за рахунок розробки моделей і методів криптографічного захисту інформації, що передається пристроями з обмеженими обчислювальними ресурсами.

У відповідності до такої мети, для вирішення наукового завдання, сформульовані такі часткові завдання дослідження:

1. Розробити метод криптографічного захисту інформації в мережі IoT на основі модифікації алгоритму A5/1 для забезпечення підвищеної стійкості шифрування та імітостійкості.

2. Побудувати криптографічний протокол інформаційного обміну в мережі для забезпечення безпечного формування сеансових ключів та забезпечення криптографічно захищеної передачі даних від ПООР до шлюза.

3. Побудувати модель загроз для розробки системи захисту інформації, що обробляється пристроями з обмеженими обчислювальними ресурсами в мережі IoT.

4. Дослідити ефективність методу захисту інформації на пристроях з обмеженими обчислювальними ресурсами із застосуванням модифікованого алгоритму шифрування на пристроях класу C0.

Висновки до розділу 1

1. В результаті аналізу визначено, що пристрої M2M які практично можна вважати пристроями IoT становлять значний відсоток від загальної кількості підключених пристроїв до мереж. Окрім цього, темпи розвитку не лише зберігаються, а й збільшуються, так станом на 2023 рік кількість таких пристроїв більше 14 млрд тобто близько 50% від всіх пристроїв, що підключаються до мереж. Було виявлено, що такі пристрої мають багато вразливостей через те, що часто мають обмежені обчислювальні ресурси, що ускладнює реалізацію стандартних методів захисту інформації.

2. В ході опрацювання літератури були виявлені основні вразливості систем IoT, та основні вимоги щодо захисту які ставляться перед дослідниками. Так розроблювані методи та алгоритми криптографічного захисту повинні враховувати обмеженість пристроїв у обчисленнях, тобто бути «легкими» з точки зору реалізації та невимогливими до споживаних ресурсів: пам’яті, тактів процесору, енергії.

3. В результаті аналізу існуючих алгоритмів захисту було визначено, що існує необхідність розробки методу криптографічного захисту інформації,

який би ефективно працював на пристроях класу C0, які повинні мати чіткі вказівки, щодо формування секретних параметрів та схеми управління ключами.

4. За результатами дослідження було визначено, що не існує чітких критеріїв вибору протоколів передачі даних. Також більшість з них не можуть бути реалізовані через брак обчислювальних ресурсів, а ті, що можуть бути реалізовані є незахищеними. Таким чином, це зумовлює необхідність побудови криптографічного протоколу інформаційного обміну в мережі для забезпечення криптографічно захищеної передачі даних від ПООР до шлюза.

5. Були розглянуті різноманітні методики аналізу інформаційних ризиків. В ході аналізу визначено, що розроблювана модель повинна забезпечувати достатній рівень безпеки згідно прийнятих стандартів аналізу інформаційних ризиків як ДСТУ ISO/IEC 27002:2023. та NIST SP 800-30. Окрім цього, було визначено, що реалізація такого алгоритму повинна бути ефективною з точки зору споживання енергії та обчислювальних ресурсів, а її компоненти повинні бути надійним.

6. Таким чином, вирішенню підлягає актуальне наукове завдання, сутність якого полягає в розробці моделей та методів захисту інформації, що передається відкритими каналами зв'язку пристроями з обмеженими обчислювальними ресурсами в мережах інтернету речей за допомогою криптографічних перетворень, що забезпечують підвищений рівень конфіденційності, криптостійкості, імітостійкості та високу швидкість шифрування на основі модифікації стандартного криптографічного алгоритму A5/1.

РОЗДІЛ 2 АНАЛІЗ МОДЕЛЕЙ ТА АЛГОРИТМІВ ЗАХИСТУ ДАНИХ НА ПРИСТРОЯХ З ОБМЕЖЕНИМИ ОБЧИСЛЮВАЛЬНИМИ РЕСУРСАМИ

2.1. Критерії аналізу функціонування алгоритмів на пристроях з обмеженими обчислювальними ресурсами в мережі інтернету речей

IoT – це величезна мережа пристроїв, що активно розвивається, до якої входять різноманітні пристрої від розумних автомобілів до акумуляторних мініатюрних датчиків. Ці пристрої значно відрізняються в плані обчислювальної потужності, а також щодо їх швидкості передачі даних і обсягу обчислювальних ресурсів. IoT можна розглядати як велику екосистему, що складається з високо диверсифікованих та неоднорідних пристроїв, які мають різні характеристики, розміри та вимоги до застосування. Як наслідок існує десятки різних (і в основному несумісних) мікроконтролерних платформ, операційних систем та стандартів безпроводової комунікації для IoT, багато з яких оптимізовані для обслуговування певної галузі з специфічними вимогами та обмеженнями.

Ця неоднорідність пристроїв IoT різко контрастує з «монокультурою» в мережах класичних комп'ютерів, таких як стаціонарні комп’ютери або ноутбуки, де 64-розрядні архітектури Intel та AMD становлять абсолютну більшість на ринку. Тим не менш, 64-розрядні процесори Intel та AMD представляють лише невелику частку IoT в цілому, в якому (кількісно) домінують мікроконтролери з досить незначними обчислювальними можливостями.

У відповідності до аналізу глобального ринку мікроконтролерів [71] у 2021 році ринок поділений як зображено на рис. 2.1. При цьому, загальний

ринок мікроконтролерів продовжить свій ріст із середньорічним показником в 10,10% в період з 2022 року по 2028, де ріст ринку 8-бітних контролерів буде складати 4,70%. Проте, найбільший ріст очевидно будуть демонструвати 32-бітні контролери через зниження їх ціни та вимоги в обчислювальних ресурсах.

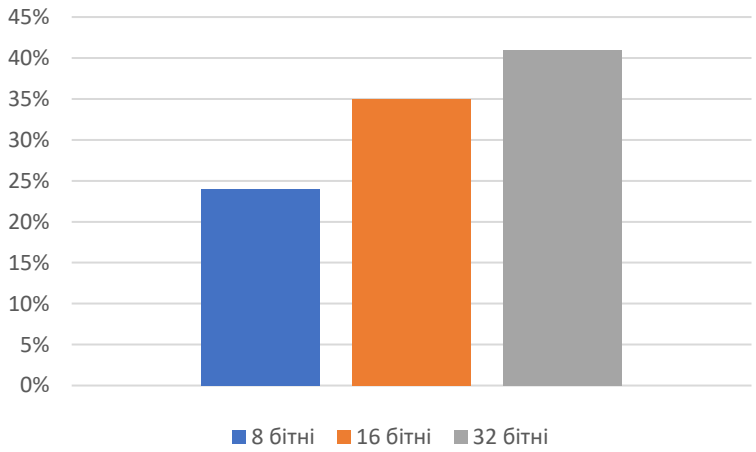

Рис. 2.1. Ринок мікроконтролерів станом на 2021 рік

Зважаючи на таке, ринок 8-бітних контролерів продовжує свій розвиток, окрім цього ці дані лише демонструють продажі мікроконтролерів, та не враховують скільки їх вже існує на сьогоднішній день. Відповідно можна зробити висновок, що 8-бітні контролери використовуються в мережах IoT в достатньо великій кількості, як наслідок існує необхідність в розробці алгоритмів та методів для захисту ПООР класу С0, через їх кількість та прогнозований ріст кількості в майбутньому.

Оскільки, в IoT немає єдиної домінуючої платформи мікроконтролерів, важливо, щоб криптографічні алгоритми досягали стабільної продуктивності на широкому спектрі архітектур 8, 16 і 32 біт. Це далеко не проста задача, оскільки, наприклад, 32-бітний мікроконтролер ARM Cortex-M3 має значні архітектурні та мікроархітектурні відмінності в характеристиках, ніж 8-бітний мікроконтролер AVR ATmega. Перший має 16 регістрів, з яких 14 доступні для загального користування, тобто простір загального призначення становить 448

біт. З іншого боку, мікроконтролери AVR мають 32 загальні регістри, але кожен з них може вмістити лише 8 біт даних, що означає, що робочий простір регістрів становить 256 біт. ARM і AVR також значно відрізняються у своїй здатності виконувати багатобітні зсуви та ротації, які є критичними для продуктивності операціями багатьох симетричних блокових шифрів. Арифметико-логічний блок мікроконтролерів ARM має швидкий ковзний перемикач, який може зсунути або повернути 32-бітне слово на довільний набір біт за один такт. Крім того, зсув або поворот можна поєднати з багатьма іншими арифметико-логічними інструкціями, що означає, що багатобітні зсуви та ротації в основному не вимагають якихось значимих обчислювальних ресурсів на ARM. Для архітектур 8 і 16 біт ситуація зовсім інша, оскільки більшість з них мають лише однобітні інструкції зсуву та повороту, що означає, що зсув або поворот регістра на n біт займає (принаймні) n + 1 такт. Таким чином, виконання зсуву або повороту 32-бітного слова на AVR, у гіршому випадку, може вимагати десятків тактів. Це автоматично робить алгоритми, що були розроблені для 64- та 32-бітних систем, менш ефективними на 8- та 16-бітних системах, через те, що окрім значної кількості ресурсів таким процесорам необхідно виконувати більше операцій.

Існують три основи КПЗВ NIST [16]:

- обчислювальна потужність (використання процесору та ОЗП, час виконання);
- живлення акумулятора (використання процесору та ОЗП, розряд акумулятора, час виконання);
- фізичний простір (використання ОЗП, еквіваленту логічних вентилів ‘Gate equivalent’ (GE), площа реалізації).

Крім цього, є також вимоги щодо ПЗП, яку використовує певний алгоритм, що пов'язано з розміром коду та його ефективністю.

Наприклад, пасивні RFID-пристрої не мають батареї для живлення, і чіп повинен живитися від енергії, отриманої від радіохвилі. Таким чином, RFID-пристрій, ймовірно, буде істотно обмежений розрядом батареї, пов'язаним з

будь-якими криптографічними функціями, а також обмежений вимогами до часу і кількості GE. Навіть якщо, RFID-пристрій має батарею (активний RFID), може бути важко зарядити батарею, тому розряд батареї часто потрібно мінімізувати.

Тому часто існує компроміс між використовуваним методом криптографії та загальною безпекою методу. Таким чином, часто методи КПЗВ NIST балансують продуктивність, в тому числі швидкість з розрядом батареї та розміром реалізації або GE та можуть мати нижчий рівень криптостійкості на відміну від стандартних криптографічних алгоритмів (як AES і SHA-256). Крім цього, метод повинен також мати низькі вимоги до ОЗП (використанні під час роботи) і ПЗП (для реалізації на пристрої).

КПЗВ NIST охоплює дуже широкий спектр ПООР, алгоритми шифрування на них можуть бути реалізовані як на апаратному, так і на програмному рівні. КПЗВ NIST робить компроміси між необхідними ресурсами для реалізації, швидкістю, безпекою, продуктивністю та енергоспоживанням на ПООР. Основна ідея КПЗВ NIST полягає в тому, щоб використовувати менше пам'яті, менше обчислювальних ресурсів та менше енергії для забезпечення достатнього рівня захисту. КПЗВ NIST повинні бути простішими та швидшими порівняно з класичною криптографією. Проте недоліком КПЗВ NIST є менший рівень безпеки [61].

При апаратній реалізації КПЗВ NIST важливими метриками є розмір реалізації в GE, споживання ОЗП та ПЗП та споживання енергії. Для зваженої оцінки КПЗВ NIST слід враховувати: точний тип схеми, пам'ять, зберігання внутрішніх станів та станів ключа. Однак це не означає, що коротші розміри блоку та ключа є кращими, оскільки це може призвести до вразливостей проти атак на зв’язаних ключах [61]. У деяких випадках використовується технологія тільки читання, щоб знищити ключі в пристрої (чипі), для зменшення простору ключів. В [8] запропоновано метрика енергоефективності апаратної реалізації, в якій затримка використовується для оцінки часу, необхідного для виконання заданої операції [6].

Для програмних реалізацій важливими метриками для КПЗВ NIST є розмір реалізації та споживання ОЗП, а також швидкість шифрування (байти на цикл). Окрім цього, важливим залишається якомога менше споживання енергії.

Варто зауважити, що окрім ефективності з точки зору швидкості та використанні мінімуму обчислювальних ресурсів, важливим є рівень криптографічного захисту певного алгоритму, зокрема імітостійкість шифротексту.

Зважаючи на таке, а також проаналізувавши інші подібні дослідницькі роботи [89, 66], доцільність застосування тих чи інших методів, криптографічних алгоритмів або протоколів, що адаптовані для реалізації в середовищах пристроїв на яких не можуть бути реалізовані стандарти безпеки, розроблені для традиційних комп'ютерних мереж, можна визначити за наступними показниками.

Вимоги до енергії

Кількість енергії, що вимагається схемою для обробки алгоритму, може вимірюватися в мікроватах або в джоулях за секунду.

Споживання енергії на біт

Споживання енергії на біт можна розрахувати наступним чином [34]:

$$E = \frac{d * W}{S}, \tag{2.1}$$

де d – затримка в циклах/блок;

W – потужність в мкВт;

S – розмір блоку в бітах.

В (2.1) затримка використовується в термінах програмної реалізації.

Значення розміру ключа, розміру блоку тексту, вимог до оперативної та постійної пам'яті, площі реалізації, затримки, швидкості шифрування визначаються наступним чином:

Швидкість шифрування в апаратному забезпеченні, може вимірюватися в термінах обробленого тексту за одиницю часу (біт за секунду). Швидкість в бітах за секунду може бути обчислена за формулою:

$$V = \frac{Ps}{t},$$

де Ps – розмір тексту в бітах;

t – витрачений час.

Розмір реалізації (для апаратної реалізації)

Цей простір може бути задано за допомогою логічних блоків для програмованих користувачем вентильних матриць або за допомогою GE для інтегральних схем (1GE = 2 вхідний логічний вентиль I-HE). Зазвичай, 200–2000 GE (з 1000–10 000 GE загальної доступної) виділяються для цілей безпеки в мітці RFID [44].

Затримка – це час, необхідний для отримання шифру з оригінального тексту з точки зору продуктивності апаратного забезпечення [58], тоді як кількість тактів на блок (під час шифрування) визначає затримку програмного забезпечення. Затримку швидкості шифрування або розшифрування можна обчислити використовуючи дані отримані з обчислення пропускної здатності за формулою:

$$d = \frac{f}{\frac{Ps}{t}/Bs} = \frac{f*Bs*t}{Ps},$$

де f – частота процесора в герцах;

Bs – розмір блоку в байтах.

Вимоги до пам'яті, зазвичай, вимірюється в кБ [34]. ОЗП потрібно для зберігання проміжних значень, які можуть використовуватися в обчисленнях, а ПЗП потрібно для зберігання програми/алгоритму та статичних даних, таких як ключ алгоритму, S-блок (якщо використовується) тощо [58]. З урахуванням достатньо великого розриву між наявною пам’яттю на ПООР, навіть в межах одного класу, доцільно визначити допустимим використання 20% пам’яті пристрою на реалізацію алгоритму шифрування, по аналогії з апаратною реалізацією. З точки зору використання оперативної пам’яті, допустимий

обсяг використання в цілях безпеки, може бути різним в залежності від обчислювальних завдань системи.

Таким чином для визначення недоліків та переваг існуючих рішень для захисту обмежених пристроїв, необхідно провести тестування за визначеними параметрами. Проте при проведені аналізу та оцінки, необхідно враховувати, що допускається компроміс між ефективністю та рівнем захисту, тобто ефективність має більший пріоритет [64].

2.2. Дослідження існуючих алгоритмів захисту даних для пристроїв класу C0

Дослідження і розробка методів КПЗВ NIST для використання на ПООР в мережах IoT активно розвивалися протягом останнього десятиліття. Основною метою є створення і використання невибагливих до обчислювальних ресурсів криптографічних алгоритмів, які можуть бути застосовані такими пристроями, забезпечуючи при цьому належний рівень криптографічного захисту. Для дослідження було відібрано декілька сучасних та відомих алгоритмів КПЗВ NIST, для визначення їх ефективності функціонування на пристроях класу C0, з 8- та 16-бітними процесорами.

PRESENT – це блочний шифр, орієнтований на апаратне забезпечення, вперше запропонований у 2007 році на конференції з криптографічного обладнання та вбудованих систем. PRESENT вибраний до аналізу через його швидку взаємодію та через те, що 2012 році організація ISO та IEC визначили алгоритми PRESENT та CLEFIA у міжнародному стандарті обмеження доступу ISO/IEC 29192-2:2012 [41]. Автори стверджують, що він має компактний розмір апаратної реалізації (1570 GE), і призначений для пасивних міток RFID. PRESENT має 80-бітний або 128-бітний ключ шифрування. Він використовує розмір блоку 64 біта і має 16-розрядний шар дрібно-нелінійної

заміни. Шар p-блок – це вузол перестановки бітів. Частини ключа шифрування використовуються операцією XOR протягом 32 раундів. Процес дешифрування є зворотним до процесу шифрування, так само, як і S-блок (4x4) і p-шар.

Автори стверджують, що PRESENT є гарним варіантом для пристроїв IoT з 8- або 16-бітними процесорами.

Кожен раунд алгоритму складається з трьох етапів: додавання ключа, нелінійного підстановочного шару та побітового перестановочного шару.

На першому етапі, який називається додавання ключа, поточний стан шифру об'єднується з підключем раунду за допомогою побітового XOR. Підключі раунду є частиною секретного ключа шифру.

На другому етапі, який називається нелінійним підстановочним шаром, результат етапу додавання ключа проходить через S-блок. S-блок є 4-бітним тензором, який застосовується 16 разів паралельно.

На третьому етапі, який називається побітовим перестановочним шаром, вихід нелінійного підстановочного шару перебудовується за допомогою перестановки. Перестановка є бітовою операцією, яка змінює порядок бітів у виході нелінійного підстановочного шару.

Ці три етапи повторюються 32 рази, утворюючи 32 раунди шифру PRESENT.

Проте, криптографічний алгоритм PRESENT відомий такими слабкостями, як атаки на сторонні канали та проблеми з витіканням пам'яті, подібні до AES. У статті [56] автори проводять аналіз вразливостей проти програмно-впровадженого PRESENT, їхні результати демонструють, що функція додавання ключа раунду вразлива до витоку інформації.

PRINCE [13] – це блочний шифр з малими вимогами до обчислювальних ресурсів, опублікований на конференції Asiacrypt 2012 року. Він оптимізований для низької затримки при реалізації в апаратному забезпеченні. Побудований на основі конструкції, подібної до конструкції Івен-Мансур (так званої конструкції FX [12]), і має цікаву особливість: розшифровування можна

виконати, повторно використовуючи процес шифрування з дещо іншим ключем. Ця особливість, так звана властивість α-відображення, дає явну перевагу в реалізаціях, що вимагають як шифрування, так і розшифровування. Проте такий підхід зменшує рівень безпеки порівняно з ідеальним шифром, таким чином автори заявили, що безпека шифру гарантується до 2^{127}-n операцій, коли проводиться 2n запитів на шифрування/розшифровування. Це обмеження є дійсним лише для моделі з одним ключем, і автори не зробили жодних заяв щодо моделі пов'язаного ключа.

Шифр має 12 раундів у своїй основі, і кожна функція раунду складається з додавання константи, що залежить від раунду і фіксованого ключа, 16 паралельних S-блоків і лінійної дифузії. Перша половина 128-бітного секретного ключа використовується як ключ перед-відбілення і після-відбілення, тоді як друга половина ключа використовується безпосередньо в функціях раунду. Для дешифрування ключ спочатку об’єднується операцією XOR з фіксованим значенням, і ту саму схему можна повторно використовувати для шифрування. Таким чином, мінімізуються витрати на дешифрування. Перед-відбілення і після-відбілення, є важливими компонентами алгоритму шифрування PRINCE. Перед-відбілення додає до даних шифру частину ключа, а після-відбілення видаляє цю частину ключа.

Функції раунду R і RG1 є основними компонентами алгоритму шифрування PRINCE. Вони складаються з XOR з фіксованим ключем, XOR з round-залежною константою, шару S-блоків і лінійною дифузією M.

Залежна від раунду константа RC використовується для забезпечення різноманітності в операціях раунду. Шар S-блоку S є нелінійною функцією, яка робить шифр більш стійким до атак.

Проте PRINCE має деякі вразливості, зокрема в [42] автори дослідили, що PRINCE не є стійким до атак на пов’язані ключі, і що можна відновити секретний ключ за допомогою однієї пари пов'язаних ключів, окрім цього вони зробили висновок, що вибір значення α є насправді чутливим для безпеки шифру.

Ще один шифр який варто розглянути з огляду на його швидкодію на ПООР – *HIGHT* [36]. Це «ультралегкий» алгоритм, який обробляє 64-розрядний блок з 128-бітним ключем за 32 раунди, за допомогою компактної функції раунду (без S-блоків) і простих обчислювальних операцій. Найкомпактніша версія використовує 2608 GE для пропускної здатності 188 кБіт/с [55].

HIGHT складається з 4 основних блоків: таблиці ключа, початкової трансформації, 32 ітераційної операції раунду та фінальної трансформації. Він шифрує 64-бітний відкритий текст у відповідний 64-бітний шифротекст за допомогою 128-бітного основного ключа використовуючи нерівноважну мережу Файстеля з 8 гілок на кожному раунді. Планування ключів може виконуватися паралельно з функцією раунду.

Процес планування ключів відповідає за генерацію ключів відбілювання та підключів для всіх блоків раундів за допомогою 128-бітних головних ключів. Він генерує вісім ключів відбілювання від WK0 до WK7. WK0 до WK3 і WK4 до WK7 передаються в початкову та фінальну трансформації відповідно. Крім того, планувальник ключів генерує 128 підключів від SK0 до SK127 для 32 ітерацій раундів. Він передає чотири підключі до кожної функції раунду.

Початкова та фінальна трансформації використовують ключі відбілювання, які приховують інформацію, що використовується для внутрішніх операцій. Ці ключі відбілювання та підключі раунду отримуються шляхом перемішування вхідного головного ключа та констант, що генеруються лінійним зворотним зсувним регістром (РЛЗЗ).

Три етапи шифрування HIGHT такі: початкова трансформація, 32 повторення функції раунду та фінальна трансформація. Процес шифрування HIGHT починається з початкової трансформації, яка обробляє 64-бітний відкритий текст чотирма ключами відбілювання. Він виконує 32 ітерації раунду: кожна з них використовує чотири підключі. Вихідні значення одного раунду стають вхідними значеннями наступного раунду. Фінальна

трансформація застосовується до результату останнього раунду разом з іншими чотирма ключами відбілювання. В результаті 64-бітний шифрований текст є результатом фінальної трансформації.

На HIGHT були застосовані різні атаки, такі як диференціальна атака на 26 раундів [69], атака на зв’язаних ключах на повний раунд [47], атака по повному дводольному графу на повній версії раунду [86] та атаки з нульовою кореляцією на 26- і 27-раундовий шифр [93]. Алгоритм HIGHT не пройшов до конкурсу NIST через вразливості в безпеці, проте він був відібраний для тесту через дуже високу швидкодію на обмежених пристроях та через те, що 29 грудня 2006 року був затверджений стандартним алгоритмом шифрування в Південній Кореї [88] Телекомунікаційним Технологічним Об'єднанням, Південна Корея, за номером стандартизації TTAS.KO-12.0040.

Ще один шифр фіналіст конкурсу NIST – *Isap* – це сімейство шифрів з автентифікацією на основі одноразового ключа з асоційованими даними, розроблених з акцентом на стійкість до пасивних атак на бічні канали [22]. Усі члени сімейства Isap є алгоритмами, що базуються на перестановці, які поєднують варіанти заснованого на губці режиму Isap з одним із декількох опублікованих перестановок.

Основною метою дизайну Isap є забезпечення готової до використання стійкості до певних типів атак на реалізацію, при цьому дозволяючи додавати додаткові механізми захисту за низькою ціною. Це є істотним у всіх випадках, коли криптографічні пристрої розгортаються в місцях, які фізично доступні потенційним зловмисникам.

Режим роботи Isap був опублікований на FSE 2017 як підхід для забезпечення вродженої безпеки проти атак диференціального аналізу потужності. Такі атаки становлять найпотужніший клас пасивних атак по бічному каналу на практиці і працюють шляхом накопичення інформації про секретний ключ шляхом спостереження за кількома шифруваннями (або дешифруваннями) різних вхідних даних. Шляхом інтеграції функції переключення на основі губки в конструкцію шифрування, завжди

використовуються свіжі ключі для обробки нових даних, Isap значно підвищує стійкість до атак DPA та пов'язаних з ними атак.

Автори стверджують, що всі члени ISAP забезпечують 128-бітну безпеку, і ISAP спеціально розроблений для забезпечення безпеки проти атак по бічному каналу. Вони також зазначають, що функція зміни ключів запобігає диференціальним атакам на потужність та атакам з помилками.

Специфіка структури на основі губки допомагає підвищити опір простим атакам на потужність. Однак криптографічний нонс ніколи не повинен використовуватися повторно з тим самим вільним текстом.

Як стверджують автори, дизайн ISAP підходить для легкої реалізації в програмному та апаратному забезпеченні. Апаратна реалізація ISAP-K-128A потребує лише площі в 12 kGE в бібліотеці осередків TSMC65nm. Програмна реалізація ISAP-A-128A використовує 450 циклів на байт на мікроконтролері AVR ATmega328p [21]. Обидві апаратні та програмні реалізації забезпечують надійну безпеку проти певних атак з помилками, включаючи DFA, SFA та SIFA.

Для аналізу також було обрано відомий алгоритм Extended Tiny Encryption, відомий як XTEA – алгоритм шифрування невимогливий до обчислюваних ресурсів, побудований на основі 64-бітної блокової мережі Файстеля, алгоритм шифрування XTEA був розроблений з оригінального TEA тими ж авторами як розширення, в якому він був вказаний як цінна та інноваційна альтернатива для підвищення безпеки, як доповнений операціями перемішування ключів. Його невеликий розмір коду, низькі вимоги до обчислювальних ресурсів та простота у реалізації дозволяють використовувати його для операцій шифрування програмного забезпечення, які зазвичай розміщуються на невеликих вбудованих системах [113, 83].

Хоч XTEA вважається достатньо відомим алгоритмом серед тих, що споживають мало обчислювальних ресурсів, він забезпечує нижчий рівень безпеки, у випадку використання невеликої кількості раундів, тому повинен мати можливість вмістити 32 раунди, щоб задовольнити потреби додатків з високою безпекою [37].

XTEA реалізує шифрування за допомогою 64-бітного блоку, розділеного на два 32-бітних півблоки, v0 і v1, які вводяться в алгоритм, він виконує 32 раунди (Nr = 32). У XTEA схема керування ключами модифікується, щоб відобразити різні шаблони для безперервного змішування даних і ключа в кожному раунді.

Використовуються лише чотири підключі, кожен довжиною 32 біт, і базові операції додавання та віднімання за модулем 2^{32}. Логічні зсуви – це бітовий зсув вліво на чотири та бітовий зсув вправо на п'ять, а також проста 32-бітна операція XOR [94].

Функції перестановки виражаються як $f(x) = (x << 4 \oplus x >> 5) + x$, а функції генерації підключів виражаються як $sum + k\,(sum \vee 3)$ і $sum + k\,(sum >> 11 \vee 3)$. Сума діє як селективний елемент з чотирьох підключів k0, k1, k2 і k3, залежно від бітів 0 і 1 суми або бітів 11 і 12. Результати функції перестановки та згенеровані підключі XOR і ADD додаються до v0 і v1. Варто зазначити, що значення суми ініціалізується до нуля перед початком обчислення, а значення delta фіксується як 0x9E3779B9.

Також було відібрано дуже важливий для аналізу алгоритм Ascon, через його перемогу в конкурсі від NIST, та плани NIST щодо стандартизації його як рекомендованого алгоритму для пристроїв з обмеженими обчислювальними ресурсами.

Принцип його роботи побудований на конструкції губки розміром 320 біт (що складається з п’яти 64-бітних слів x0, . . . , x4). Ascon існує в двох варіантах, Ascon128 і Ascon-96, з різними рівнями безпеки та параметрами. Для аналізу був обраний саме алгоритм Ascon-128.

Режим роботи Ascon відбувається за режимом аутентифіковане шифрування з приєднаними даними. Це означає, що повідомлення спочатку шифрується, а потім перевіряється цілісність. Перестановка Ascon – це складний алгоритм, який використовує ряд операцій для змішування даних і ключа. Це робить Ascon більш стійким до атак.

Режим роботи Ascon заснований на архітектурі MonkeyDuplex [10]. Процес шифрування розділений на чотири фази:

– ініціалізація;
– обробка асоційованих даних;
– обробка відкритого тексту;
– завершення.

Ці фази використовують дві різні перестановки p^a і p^b. Варіант p^a використовується для ініціалізації та завершення, тоді як p^b використовується в фазах обробки даних.

Ініціалізація приймає в якості входу секретний ключ K і відкритий криптографічний нонс N. Ініціалізація гарантує, що алгоритм починає з випадкового стану на початку фази обробки даних для кожного нового криптографічний нонс . У подальшій обробці асоційованих даних блоки r-бітів поглинаються шляхом їх згортання з байтами стану, розділяючи їх викликами p^b. Якщо не потрібно обробляти асоційовані дані, то всю фазу можна пропустити. Відкритий текст обробляється в блоках r-бітів у подібний спосіб, з блоками шифротексту, витягнутими з стану відразу після додавання відкритого тексту. Для розділення доменів між асоційованими даними та відкритого тексту до секретної частини внутрішнього стану додається константа. Після обробки всіх даних алгоритм повертає k-бітну мітку T.

Ascon використовує дві перестановки p^a і p^b. Обидві ітеративно застосовують ту ж саму функцію раунду p: a раундів для p^a, і b раундів для p^b. Кругова трансформація p складається з додавання константи до x2, за якою слідує застосування нелінійного шару заміни та лінійного шару.

Шар заміни використовує 5-бітний S-блок, який є аффінно еквівалентним χ-відображенню Keccak [11]. S-блок Ascon застосовується 64 рази паралельно. Кожен біт з п’яти 64-бітних слів (x0, ..., x4) вносить один біт у кожен з 64 S-блок, де x0 завжди служить найзначущим бітом.

Лінійний шар походить від Σ-функції SHA-2 [62]. Функція Σ застосовується до кожного з 5 слів стану і використовує різні значення повороту для кожного слова.

Таким чином, необхідно оцінити рівень ефективності відібраних алгоритмів на пристроях класу C0 за визначеними критеріями, для визначення реальних можливостей та перспективи застосування.

2.3. Оцінка функціонування існуючих алгоритмів на пристроях з обмеженими обчислювальними ресурсами

Ціллю даного аналізу є визначення ефективності роботи найвідоміших алгоритмів легкого шифрування на пристроях класу C0. Необхідність проведення аналізу зумовлена тим, що в ряді досліджень зі схожими параметрами системи, дослідниками отримані суперечливі результати [26, 66, 89]. Для реалізації алгоритмів обрано мікроконтролер ATMEGA328p на платформі Arduino (рис. 2.2). Це дуже розповсюджений контролер для навчання, прототипування та комерційного застосування. Його перевагами є простота в програмуванні та відкритість платформи. Він програмується на мові C з додатковими програмними блоками для керування периферією платформи.

Рис. 2.2. Загальний вигляд платформи Arduino Uno на базі мікроконтролеру ATMEGA328p

ATmega328P – це однокристальний мікроконтролер сімейства megaAVR з архітектурою 8-бітного процесора RISC, що працює на частоті 16 МГц. Має 32 кБ ПЗП і 2 кБ ОЗП. Такі характеристики повною мірою відповідають класу пристроїв 0, тому є гарним варіантом для проведення оцінки роботи алгоритмів. Живлення контролеру для всіх тестів крім енергоефективності було від USB 3.0 порту. Для проведення тесту на енергоефективність мікроконтролер було заживлено від лабораторного джерела живлення з напругою 7 вольт постійного струму через спеціальний інтерфейс живлення на платі.

Для проведення експерименту були обрані «найлегші» реалізації шифрів, в яких залишено лише частину коду з шифруванням та розшифруванням та змінні необхідні для роботи. Алгоритми реалізовані мовою С.

Як частина функціоналу алгоритму, сучасні шифри збільшують їхню безпеку (змішування та дифузія) за допомогою повторного виконання (n разів) простої функції раунду. У блочних шифрах вхід і вихід функції раунду рівні розміру блоку шифру в загальному випадку. Як стандартне правило, збільшення кількості раундів n підвищує рівень безпеки, тоді як зменшення кількості раундів відіграє значну роль у скороченні часу виконання

шифрування і розшифрування, що є однією з сутностей КПЗВ NIST, проте не завжди, інколи ефективнішим буде більша кількість раундів при меншій кількості операцій як в алгоритмі XTEA. Ще один алгоритм, який варто відзначити перед тестуванням Isap – в його різних модифікаціях використовується різна кількість раундів, в даному випадку для тестування була взята модифікація з 32 раундами. Що стосується ключа всі алгоритми використовують 128-бітний ключ, окрім алгоритму PRESENT де є можливість використання ключа 80 біт, проте зважаючи на вимоги NIST, розглядалась саме модифікація з розміром ключа 128 біт. Ascon та Isap мають можливість шифрувати більші блоки тексту, проте, шифрування малих повідомлень вимагає навпаки більшої кількості операцій. В табл. 2.1 наведено порівняння алгоритмів за розміром ключа, блоку та кількості раундів для забезпечення необхідного рівня захисту.

Таблиця 2.1

Порівняння алгоритмів за розміром ключа, блоку, кількості раундів

Алгоритм	Ключ, біт	Блок, біт	Раундів
PRESENT	80-128	64	32
PRINCE	128	64	12
HIGHT	128	64	32
Isap	128	64-144	32/33/48/56
XTEA	128	64	64
Ascon	128	64-128	30/32

Вимоги до пам'яті

Мікроконтролер ATMEGA328p має 32 кБ енергонезалежної пам’яті яка в основному використовується для зберігання програм. Вимірювання зайнятої програмою місця в пам’яті відбувалось за допомогою вбудованих функцій середовища програмування Arduino ide. Для тестування використовувались

мінімальні реалізації алгоритмів, тільки функції шифрування/розшифрування та всі необхідні параметри для їх роботи, результат зображено на рис. 2.3. Таким чином, більше всього місця в пам’яті займає реалізація шифру PRESENT, найменше HIGHT та XTEA. Проте варто зауважити, що розмір програм не перевищує 10% від доступної пам’яті, що є прийнятним. Окрім цього за класифікацією до пристроїв класу C0 належать пристрої в яких об’єм ПЗП пам’яті до 100 кб, тому всі алгоритми споживають не багато пам’яті залишаючи достатньо місця для корисного навантаження.

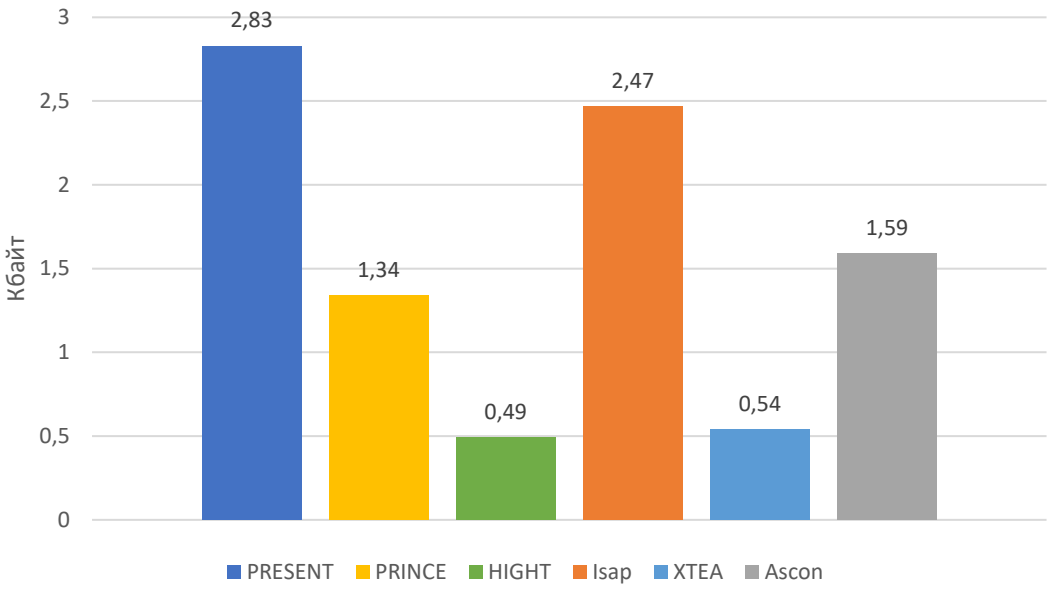

Рис. 2.3 Використання постійної пам’яті

Під час аналізу споживання ОЗП пам’яті виявлено, що найбільше оперативної пам’яті використовують алгоритми HIGHT та Isap (рис. 2.4). Оперативна пам’ять є важливим парметром, оскільки в ній зберігаються всі змінні в таких структурах як стек та куча. Пам’ять на мікроконтролерах є критичною ланкою, оскільки якщо в процесі роботи відбудеться її переповнення, наприклад, через великі масиви даних, програмний код може працювати непередбачувано, що поставить під загрозу не тільки безпеку, а і правильне виконання команд, що в деяких системах може бути критично. На платформі ATMEGA328p доступно 2 кБ оперативної пам’яті. Таким чином,

найефективнішим алгоритмом за цим показником є XTEA. Проте шифр Ascon, що планується до стандартизації NIST також демонструє непогані показники використання ОЗП. Також варто зазначити, що всі алгоритми використовують досить невеликий у відсотковому відношенні до доступного, об'єм оперативної пам'яті [120].

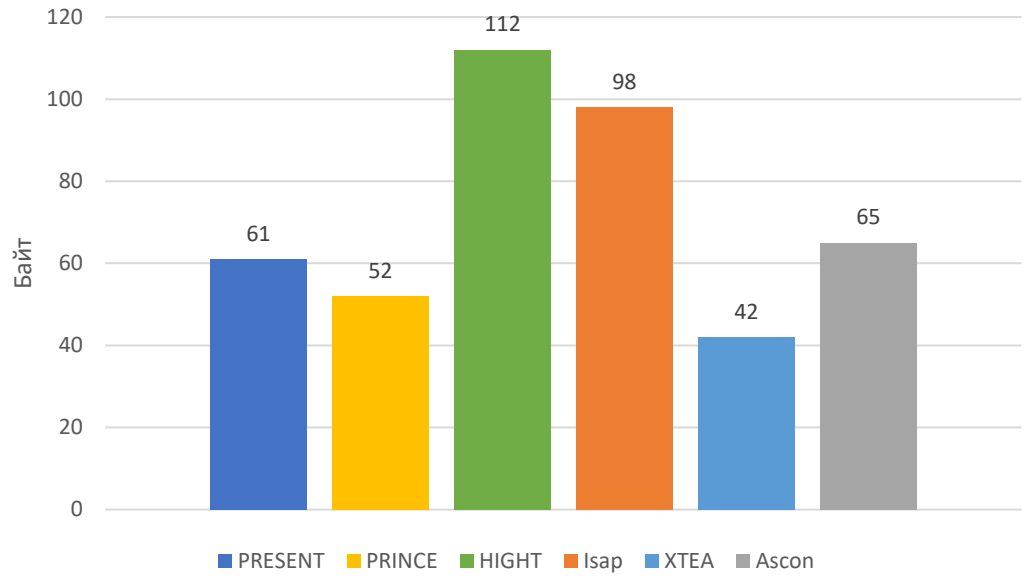

Рис. 2.4 Використання оперативної пам'яті

Показник швидкості шифрування (рис. 2.5), тобто те скільки даних може бути зашифровано за одну секунду, дозволяють побачити реальну ефективність алгоритму. Найшвидшим алгоритмом виявився алгоритм HIGHT зі швидкістю 67,1 кБ/с, проте важливо відмітити, що при цьому він і споживає найбільше оперативної пам'яті. Стабільний показник має алгоритм шифрування XTEA при мінімальному споживанні оперативної пам'яті та при малому розмірі реалізації (серед обраних до аналізу алгоритмів), він здатний підтримувати швидкість шифрування – 25,99 кБ/с. Проте такі шифри як Ascon та Isap, показали дуже низьку швидкість, це зумовлено тим, що вони розроблялись для 64-бітних пристроїв, та мають реально малу продуктивність на 8-бітних та 16-бітних пристроях класу C0, яким є ATMEGA328p.

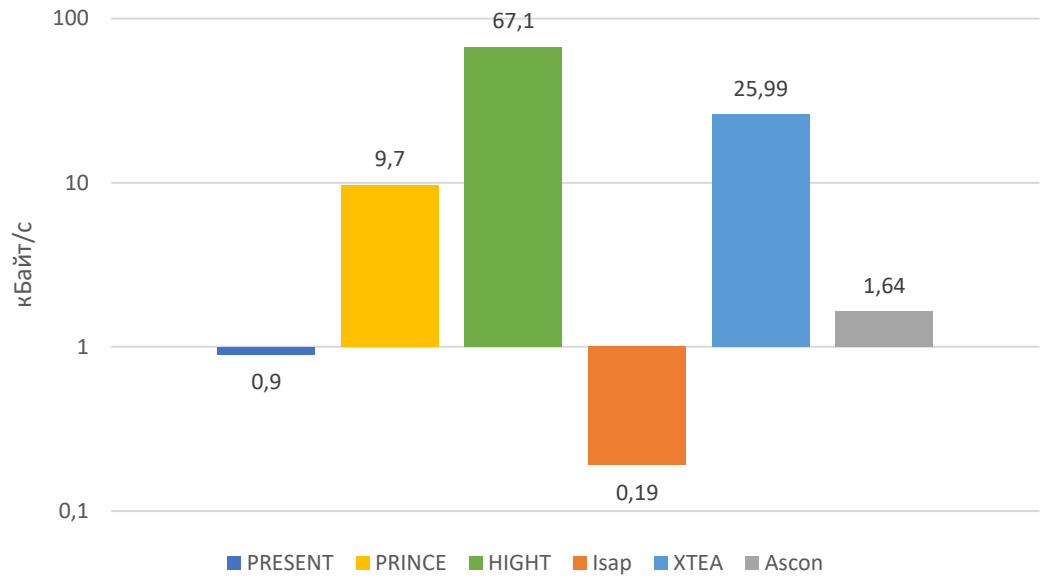

Рис.2.5 Швидкість шифрування

Програмна затримка (рис. 2.6) виглядає «дзеркально» до швидкості шифрування, адже визначає час необхідний для шифрування блока даних, з поточною швидкістю та частотою процесора. Через це алгоритм HIGHT має найкращий результат, при цьому найстабільніший при споживанні ресурсів та швидкості роботи залишається XTEA.

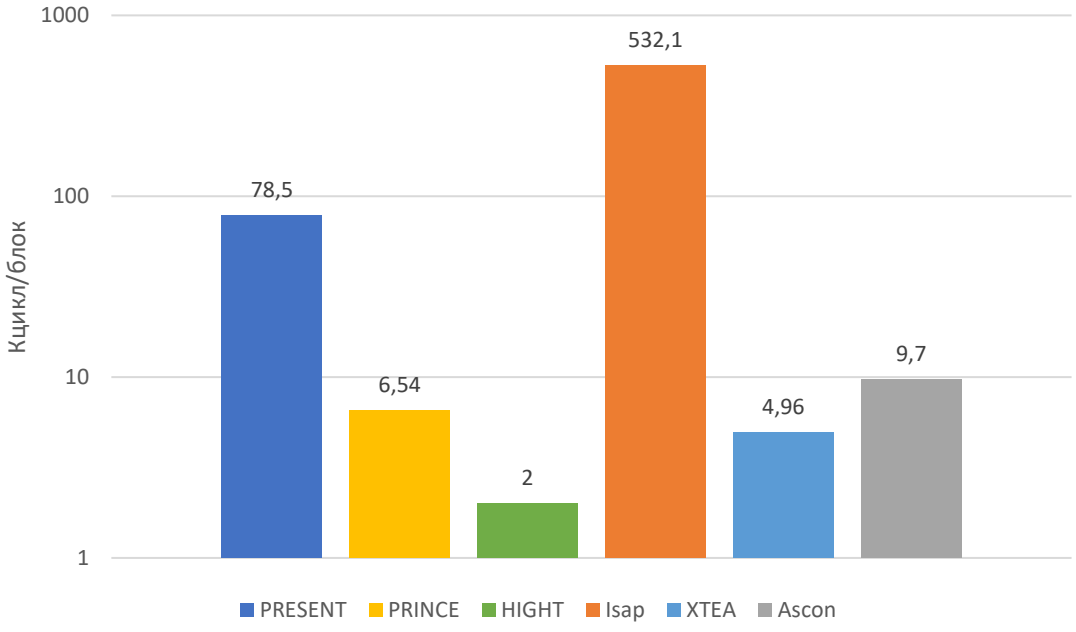

Рис. 2.6 Програмна затримка

Аналізуючи кількість енергії, що споживається пристроями (рис. 2.7), можна зробити висновок що алгоритми Isap та PRESENT споживають надто велику кількість енергії в порівнянні з іншими, при цьому вони демонструють достатньо низьку швидкість на пристроях класу C0. Такий висновок можна зробити і щодо алгоритму Ascon, кількість спожитої ним енергії є значною по відношенню до низької швидкості шифрування. Найменшу кількість енергії споживає алгоритм HIGHT. Кількість споживання енергії є критичним показником в деяких системах IoT які живляться від акумулятора.

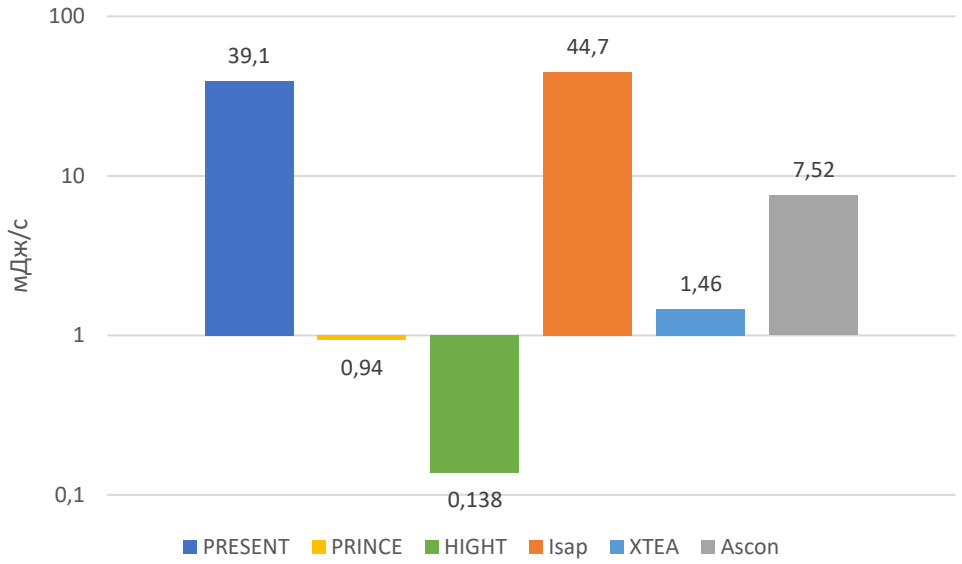

Рис. 2.7 Споживання енергії за секунду

Якщо аналізувати з точки зору спожитої енергії на біт (рис. 2.8) результат приблизно такий самий. Алгоритми Isap та PRESENT демонструють надто велике споживання як для швидкості з якою вони шифрують інформацію. При цьому HIGHT демонструє кращий результат через високу швидкість шифрування. Алгоритм XTEA демонструє збалансовану ефективність враховуючи вимогу до ресурсів та швидкість.

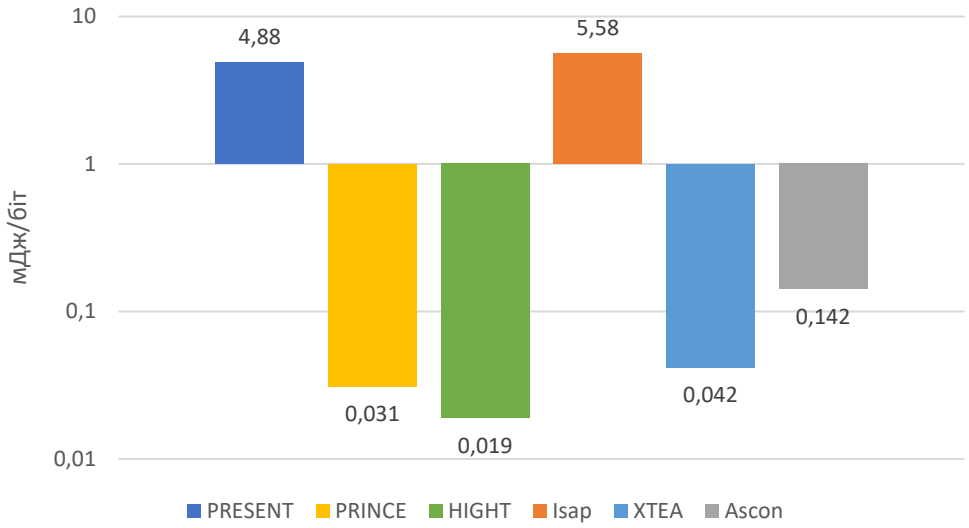

Рис. 2.8 Споживання енергії на шифрування одного біту

Відповідно до аналізу найефективнішим алгоритмом з точки зору швидкості шифрування є HIGHT, проте він вимагає значного обсягу оперативної пам’яті у порівнянні з рештою алгоритмів. Окрім цього він має деякі вразливості. Найбільш збалансованим алгоритмом за всіма параметрами можна визнати XTEA при малому розмірі реалізації та споживання пам’яті він демонструє гарний показник швидкості. Алгоритм PRINCE демонструє значно нижчі показники ефективності в порівнянні з XTEA при майже схожих вимогах до ресурсів. Що стосується алгоритмів фіналіста конкурсу NIST Isap, та переможця, що планується до стандартизації Ascon, вони демонструють низькі показники ефективності на 8-бітних пристроях з обмеженими обчислювальними ресурсами класу C0, через те, що при їх розробці в якості цільової платформи були визначені 64-бітні процесори. Щодо алгоритму PRESENT він є надто неефективним через значні вимоги до обчислювальних ресурсів і надто малу швидкість шифрування. Таким чином розроблюваний метод необхідно порівнювати з алгоритмами HIGHT – який найефективніший, XTEA – найзбалансованіший та Ascon – переможець конкурсу NIST.

2.4. Побудова моделі загроз безпеки інформації в мережі інтернету речей

Інформаційні ресурси організацій, осіб, держави мають певну цінність, яку можна виразити в матеріальному вигляді, та потребують захисту від впливів, що потенційно можуть знизити цінність таких ресурсів. В [116] визначено, що загрозою є потенційно можливий несприятливий вплив на систему.

При розробці засобів криптографічного захисту інформації (КЗІ) необхідно враховувати можливі загрози у вірогідних умовах їх експлуатації [99]. Так у відповідності до [118] на етапі створення комплексної системи захисту інформації, визначаються її мета, завдання захисту інформації, варіанти вирішення цих завдань. Після цього здійснюється аналіз ризиків, що включає вивчення моделі загроз та можливих наслідків від реалізації потенційних загроз. Вже після визначення моделі загроз визначається загальна структура комплексної системи захисту інформації та її склад.

Для розробки моделі загроз необхідно врахувати потенційні суттєві загрози та описати методи їхнього здійснення. Зокрема загрози можуть здійснюватися: технічними каналами, каналами спеціального впливу, шляхом підключення до пристроїв та каналів зв’язку [117].

В НД ТЗІ 1.4-001-2000 вказано: «загрози для інформації, що обробляється в автоматизованій системі, залежать від характеристик операційної системи, фізичного середовища, персоналу, технологій обробки та інших чинників і можуть мати об'єктивну або суб'єктивну природу. Загрози, що мають суб'єктивну природу, поділяються на випадкові (ненавмисні) та навмисні. Мають бути визначені основні види загроз для безпеки інформації, які можуть бути реалізовані стосовно автоматизованої системи і повинні враховуватись у моделі загроз» [117, с. 23].

До ненавмисних загроз відносять: випадкові дії та явища (стихійні лиха, збої системи), дії спричинені персоналом ненавмисно (неправомірна зміна режимів роботи системи, зараження комп'ютерними вірусами, невиконання вимог, встановлення забороненого ПЗ, помилки під час введення даних). До навмисних загроз, в свою чергу, відносять дії зловмисника спрямовані на порушення роботи системи і одержання доступу до секретних ресурсів, наприклад: порушення фізичної цілісності, читання залишкової інформації з пам'яті або накопичувачів, неправомірне підключення до каналів зв'язку, перехоплення, аналіз трафіку, використання засобів перехоплення випромінювань, одержання атрибутів доступу для маскування під легітимного користувача [117].

Для деталізованого переліку загроз повинно бути визначено на порушення яких властивостей спрямована загроза, джерела її виникнення та можливі способи здійснення.

Аналіз загроз безпеки інформації в мережі IoT свідчить про їх певні особливості в плані реалізації та можливих наслідків для інформації, що обробляється в системі. Звернемо увагу на типове представлення мережі IoT, загалом дослідники визначають декілька еталонних моделей. Трирівнева модель складається з мережі безпроводових сенсорів, серверів хмарних обчислень та застосунків [33]. Пятирівнева модель [5] розділяє перші два рівні попередньої моделі для спрощення взаємодії об'єктів. Семирівнева модель запропонована компанією CISCO [19], є певним розширенням двох попередніх моделей та складається з таких рівнів: фізичні пристрої, комунікаційні з'єднання, туманні обчислення, збір даних, агрегація та доступ до даних, рівень застосунків, рівень процесів та людей. Трирівнева модель є найбільш узагальнюючою, а інші моделі представляють її розширення. В будь-якому випадку верхні рівні наведених моделей можуть бути захищеними з використанням традиційних методів та алгоритмів захисту інформації. В свою чергу ПООР зазвичай використовуються в системах на першому рівні трирівневої моделі, або на трьох перших рівнях семирівневої, та

представляють з себе фізичні пристрої для збору даних та керування актуаторами в системі. В зв'язку з метою забезпечення захисту на ПООР потребують уваги саме нижні рівні функціонування мережі IoT. Таким чином модель загроз буде включати загрози які більш вірогідні для інформації, що обробляється та пересилається ПООР (рис. 1).

Рис. 2.9. Інформація в типовій системі інтернету речей

Окрім витоку інформації з обмеженим доступом, потенційний вплив від реалізованих загроз в IoT мережі може мати більш конкретний вплив на середовище, в якому така система функціонує, на відміну від традиційних мереж IT. Це випливає з відмінності між такими мережами, в IoT частіше за все є сенсори, що вимірюють фізичні показники, та актуатори які можуть впливати на фізичні характеристики певної системи [53].

Відкритим проєктом з безпеки вебзастосунків (Open Web Application Security Project, OWASP) складено список основних вразливостей в мережах IoT [68]. Варто зауважити, що в цьому списку вказані узагальнені вразливості, а не конкретні загрози, проте вони частково випливають з вразливостей. До

них відносять: слабкі та прописані в програмному забезпеченні паролі, незахищені мережеві служби, незахищені інтерфейси, відсутність безпечного механізму оновлення, використання незахищених програмних компонентів, недостатній захист конфіденційності, відсутність шифрування або контролю доступу для конфіденційних даних, відсутність захисту в налаштуваннях за замовчуванням, відсутність фізичного захисту.

Доцільно зауважити, що деякі вразливості із цього списку притаманні за замовчуванням для пристроїв класу С0, наприклад, відсутність оновлення, оскільки такі пристрої часто програмуються один раз на етапі введення в систему [14].

В основі методики побудови моделі загроз і вразливостей лежить використання експертної та статистичної інформації про загрози та вразливості притаманні досліджуваній системі [110].

Визначення ймовірності реалізації певної загрози в такому підході може бути об’єктивним і суб’єктивним. Об’єктивна ймовірність визначається як частота прояву цієї події в загальній сумі спостережень, або відношення числа прояву до їх загальної кількості. Суб’єктивна ймовірність визначається як впевненість експерта або групи експертів в тому, що подія буде відтворена [114].

По-перше, слід звернути увагу, що в мережі IoT залежно від завдань, що виконуються, потенційно може циркулювати інформація з обмеженим доступом, до якої, відповідно до законодавства можуть бути віднесені такі категорії як персональні дані, банківська та комерційна таємниця тощо. Можливо відмітити, що законодавство в сфері криптографічного захисту інформації не висуває конкретних вимог щодо реалізації відповідних механізмів і процедур.

По-друге, пристрої мережі IoT не тільки взаємодіють через потенційно небезпечне середовище, а також в багатьох випадках (системи відеоспостереження, системи сигналізації та контролю стану об’єктів критичної інфраструктури охоронні системи тощо) можуть перебувати за

межами контрольованих територій та потенційно можуть бути доступні зловмисникам.

По-третє, інформація, яка передається в мережі може бути сфальсифікована або частково модифікована із зловмисною метою.

Перелічені фактори враховані в табл. 2.2, що визначає модель загроз безпеки інформації в мережі IoT.

Таблиця 2.2

Модель загроз безпеки інформації в мережі інтернету речей

№№	Загроза	Зміст загрози	Ймовірність	Рівень шкоди
1.	Перехоплення	Отримання доступу до конфіденційної інформації	Висока	Високий
2.	Модифікація	Порушення цілісності даних що передаються	Висока	Високий
3.	Фальсифікація	Створення фіктивних даних	Середня	Високий
4.	НСД	Доступ до критичних параметрів віддалених пристроїв з метою їх спотворення або перехоплення	Висока	Високий, підвищує небезпеку реалізації інших загроз
5.	Підміна	Підміна блоків та вузлів віддалених пристроїв з метою імітації недійсної обстановки	Середня	Високий, підвищує небезпеку реалізації інших загроз
6.	Вірус	Ураження злоякісними кодами з боку мережі	Середня	Середня
7.	DDoS	Блокування роботи віддалених пристроїв з боку мережі	Середня	Середня
8.	Фізичний доступ	Отримання фізичного доступу до пристрою з метою порушення його функціональності, доступу до прошивки.	Середня	Середня

Зазначимо, що загрози №№ 1-3 можуть бути повністю або частково нейтралізовані методами криптографічного захисту інформації, загрози №№ 4-6 лише частково нейтралізуються криптографічними методами, інші з наведених загроз потребують застосування некриптографічних методів захисту.

Зокрема, в разі застосування імітостійкого потокового шифрування для захисту комунікаційного протоколу спроба вбудувати в потік даних незашифрований злоякісний код призведе до його руйнування під час розшифрування даних. При цьому ймовірність «правильного» зашифрування злоякісного коду наближується до нуля.

Водночас слід звернути особливу увагу, що реалізація загроз №№4,5 підвищує небезпеку реалізації інших загроз, що може викликати так званий «ефект доміно», що характеризує каскадне обвалення рівнів захисту.

З урахуванням запропонованої моделі загроз постає питання як комплексно розв’язати визначені проблеми та які саме криптографічні методи блокування і нейтралізації можливо застосувати в умовах обмежених обчислювальних ресурсів віддалених пристроїв мережі IoT?

Висновки до другого розділу

1. За результатами аналізу визначено, що оскільки в інтернеті речей немає єдиної домінуючої платформи мікроконтролерів, важливо, щоб криптографічні алгоритми досягали стабільної продуктивності на широкому спектрі архітектур 8, 16 і 32 біт. Було визначено ряд характеристик алгоритмів за якими можна оцінити їх продуктивність та ефективність. Серед них важливими є розмір реалізації та споживання ОЗП, швидкість, затримка, ефективне споживання енергії.

2. В ході проведеного дослідження були визначені існуючі алгоритми КПЗВ NIST котрі демонструють певний рівень захисту та високу продуктивність. Зокрема шифр PRESENT через його швидкість та через те, що він входить до стандарту ISO/IEC 29192-2:2012, PRINCE як невибагливий до ресурсів шифр, що походить від шифрів AES та PRESENT, HIGHT – через високі показники швидкодії на обмежених пристроях, Isap – фіналіст конкурсу NIST, XTEA – через його широке застосування на пристроях з обмеженими обчислювальними ресурсами та швидкодію, Ascon як переможець конкурсу NIST та майбутню стандартизацію для пристроїв з обмеженими обчислювальними ресурсами.

3. За результатами проведеної оцінки було виявлено, що запланований до стандартизації алгоритм Ascon є неефективним на пристроях класу C0, та процесорах з 8- та 16-бітною архітектурою. Найефективнішим алгоритмом серед протестованих з точки зору швидкодії є HIGHT, оскільки він забезпечує найбільшу пропускну здатність, проте він споживає більше всього ОЗП, та згідно проведених досліджень має вразливості. Найбільш збалансованим алгоритмом по співвідношенню споживання обчислювальних ресурсів до швидкості є XTEA.

4. В ході побудови моделі загроз були виявлені вразливості мереж IoT та загрози з високим рівнем потенційної шкоди та високою ймовірністю реалізації. Вказані загрози потребують нейтралізації методами криптографічного захисту інформації. Зокрема такі методи повинні забезпечувати високу криптостійкість, імітостійкість та високу швидкість шифрування. Таким чином подальшого розвитку набула модель загроз для побудови системи захисту інформації, що обробляється ПООР в мережі IoT.

РОЗДІЛ 3 РОЗРОБКА МЕТОДИКИ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ, ЩО ОБРОБЛЯЄТЬСЯ ПРИСТРОЯМИ З ОБМЕЖЕНИМИ ОБЧИСЛЮВАЛЬНИМИ РЕСУРСАМИ В МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ

3.1. Розробка методу криптографічного захисту інформації за рахунок модифікації криптографічного алгоритму A5/1 для забезпечення комунікацій пристроїв інтернету речей

При розробці методу криптографічного захисту інформації можна орієнтуватись на модель системи IoT з використанням ПООР [18] яка зображена на рис. 3.1. Дана модель виходить з класифікації ПООР [14], де вказано, що через обмеження ПООР класу С0 вони як правило обмінюються інформацією з більш потужними пристроями та не мають прямого доступу до мережі інтернет.

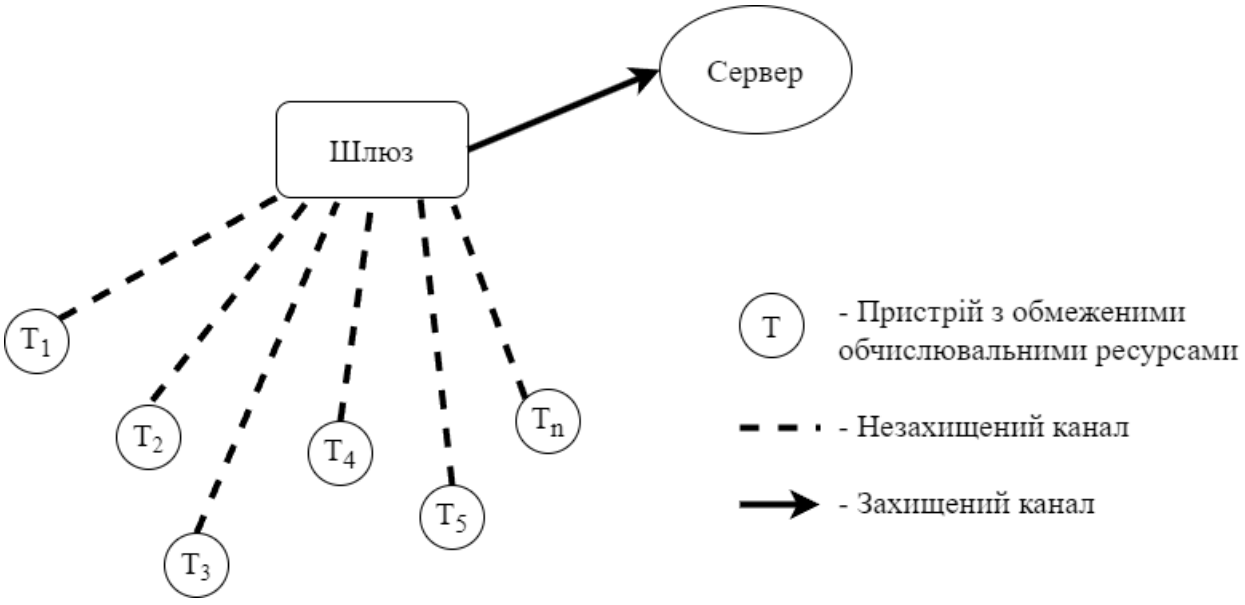

Рис. 3.1. Модель системи інтернету речей з використанням пристроїв з обмеженими обчислювальними ресурсами

Дана модель складається з:

– шлюзу для прийому даних від датчиків та передачі на сервери для подальшої обробки. В якості шлюзу використовується мікрокомп’ютер або комп’ютер з достатньою кількістю обчислювальних ресурсів для реалізації стандартних алгоритмів шифрування які можуть бути реалізовані на таких пристроях, наприклад RSA;
– пристроїв з обмеженими обчислювальними ресурсами, що використовуються в першу чергу для збору даних. Саме такі пристрої потребують алгоритмів КПЗВ NIST для шифрування даних, для безпечної передачі через незахищені канали до шлюзу як було визначено в розділі 1.2.

Також при розробці методів криптографічного захисту необхідно взяти до уваги традиційну модель системи передачі інформації з шифруванням (рис. 3.2).

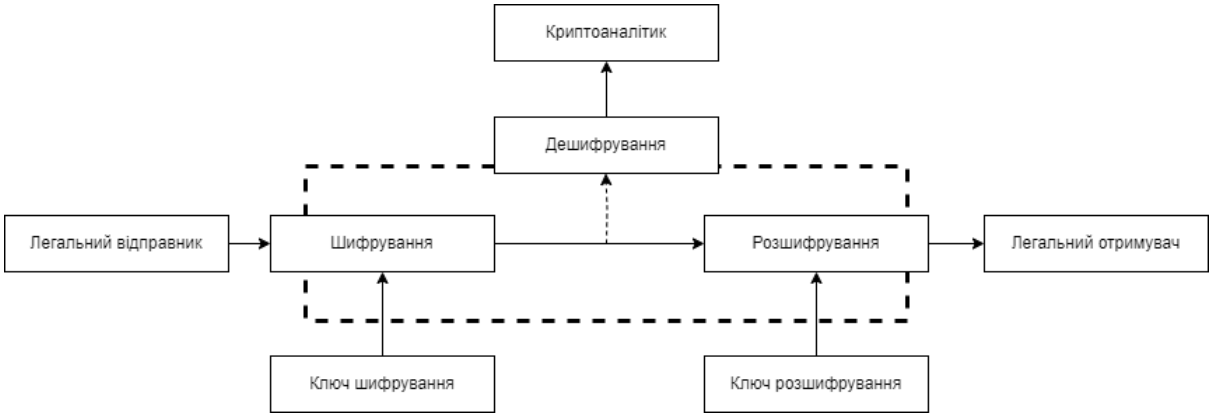

Рис. 3.2. Модель системи передачі інформації з шифруванням

Окрім цього, розроблюваний метод криптографічного захисту повинен усувати загрози визначені в складеній моделі загроз.

Переважна більшість сучасних криптографічних алгоритмів, не зважаючи на обов’язкову умову сумісності з різними програмними та апаратними платформами, що висувається під час їх проєктування [27, 20, 102,

112], може доволі неефективно працювати на пристроях з обмеженими обчислювальними ресурсами.

Модифікація сучасних алгоритмів є дуже ризикованою справою, оскільки вона може привести до прояви досі невідомих вразливостей внаслідок недостатньо досліджених їх властивостей.

Особливу категорію алгоритмів потокового шифрування становлять ті, що побудовані на основі лінійних рекурентних схем щодо яких є багато наукових-практичних досліджень в плані оцінки їхньої криптографічної стійкості та визначення потенційних вразливостей.

Зокрема, створені для забезпечення конфіденційності інформаційного обміну в радіо інтерфейсі мережі стандарту GSM алгоритм A5/1 та його дещо «послаблена» у криптографічному розумінні версія стандарту A5/2 мають багату бібліографію [17, 95] та потенціал для їх модернізації.

Слід зазначити, що згаданий вихідний криптоалгоритм A5/1 від самого початку його проєктування був націлений на забезпечення достатньо високої швидкодії (шифрування від 6400 біт/с) на ПООР, включаючи шифрування на SIM карті.

Криптосхема алгоритму A5/1 (рис. 3.3) включає три РЛЗЗ *R1*, *R2*, *R3* та схему керування рухом регістрів [17].

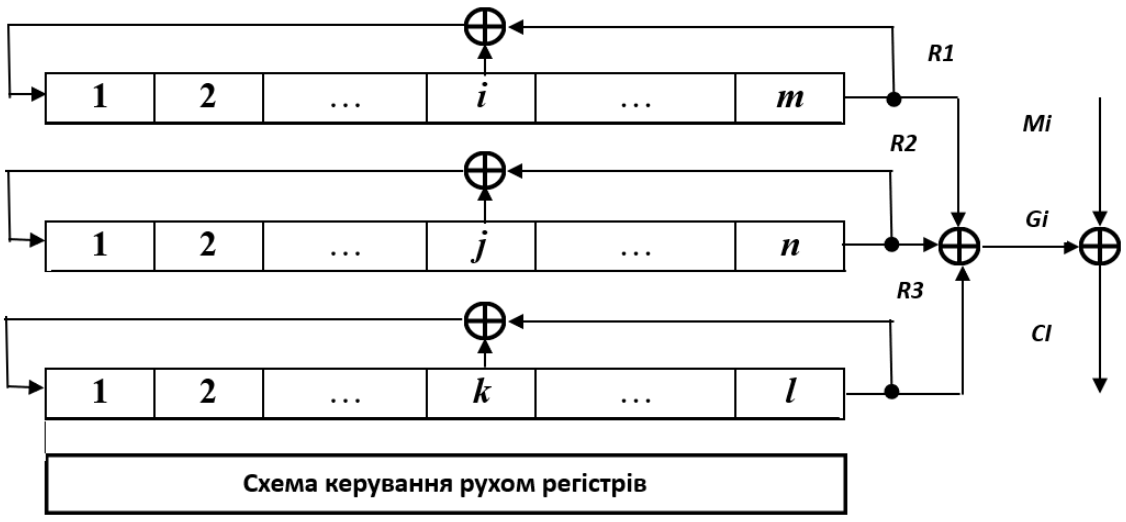

Рис. 3.3 – Криптосхема алгоритму A5/1

Довжина регістрів алгоритму A5/1 складає для *R1: m=19* бітів, *R2: n=22* біта, *R3: l=23* біта, їх початковий стан визначається сеансовим ключем, довжина якого становить 19+22+23=64 біта.

Біти зворотного зв'язку обрані таким чином, що характеристичний поліном кожного РЛЗЗ є примітивним, а це забезпечує максимальний період їх вихідних послідовностей (відповідно: $2^{19}-1,\ 2^{22}-1$ або $2^{23}-1$). Загальний період вихідної рекурентної послідовності після їх додавання за операцією XOR в випадку рівномірного руху регістрів може становити величину $\sim 10^{19}$.

Схема керування рухом регістрів в алгоритмі A5/1 реалізує їхній нерівномірний рух від такту до такту, що спрямоване на протидію криптоаналітичним атакам. Модифікований криптоалгоритм A5/2 відрізняється від A5/1 керуванням рухом трьох РЛЗЗ за допомогою четвертого РЛЗЗ невеликої довжини з рівномірним рухом, що суттєво знижує криптографічну стійкість внаслідок можливості реалізації комбінованої атаки за схемою «перебір варіантів заповнення четвертого РЛЗЗ з можливістю розв'язання систем лінійних рівнянь.

З наведеної криптосхеми (рис. 3.3) не складно бачити основні можливі недоліки A5/1 (A5/2) у випадку їх застосування для захисту інформації в мережах IoT:

- невелика довжина ключу алгоритмів утворює вразливість щодо реалізації атак типу «обмін швидкості на пам'ять»;
- вузол шифрування (операція XOR) не забезпечує достатньої імітостійкості (стійкості проти підробки) в разі проведення атак на основі відомого відкритого повідомлення;
- внаслідок відносно невеликої довжини ключу існує потенційна загроза повторення шифруючої послідовності (повторення шифру) з можливістю дешифрування відповідних відкритих повідомлень;
- алгоритм не оптимізований щодо байтової структури протоколів обміну та системи команд існуючих мікроконтролерів, що ускладнює

практичну програмну реалізацію операцій зсуву змісту регістрів на величину менш за 8, розрахунків значень функцій зворотного зв'язку тощо.

В той же час, можливо звернути увагу на те, що вказані алгоритми:

– не потребують застосування команд процесора, що призводять до надмірного енергоспоживання, зокрема, операції множення та ділення;
– на відміну від блокових криптоалгоритмів не збільшують розмір зашифрованого повідомлення, у випадку коротких/не кратних розміру блоку повідомлень.

З урахуванням нормативно визначених моделей порушника доцільно визначити мінімально необхідну довжину криптографічного ключу L виходячи з потенційно можливої обчислювальної потужності його комп'ютерної техніки W (операцій за секунду), припустимого часу на реалізацію атаки τ як:

$$\tau \leq {}^{2^L}/_{W}.$$

Звідси маємо:

$$L \geq \log_2 \tau \cdot W. \qquad (3.1)$$

Зокрема, виходячи з законодавчо визначеного терміну перегляду документів, що містять відомості кожні п'ять років та потужності сучасних суперкомп'ютерів на рівні 120-150 петафлопс з (3.1) отримуємо оцінку $L > 90$.

Остання нерівність та необхідність узгодження довжини ключу з форматом даних мікроконтролера дає підстави для визначення раціональної довжини ключу яка є $L = 128$.

Наступний крок – узгодження архітектури криптосхеми з системою команд 8-бітного мікроконтролера. Для цього кожну чарунку кожного регістра будемо розглядати як 8-ми бітове значення.

Загальна бітова довжина трьох РЛЗЗ має бути не менше довжини ключа:

$$L \leq 8m + 8n + 8l.$$

Остання нерівність дає оцінку $m + n + l \geq 16$. В той же час, необхідно розуміти, що загальний період T криптографічного перетворення не може перевищувати:

$$T \leq (2^m - 1) \cdot (2^n - 1) \cdot (2^l - 1) < 2^{m+n+l}. \quad (3.2)$$

Зауважимо, що в останній нерівності оцінка максимального періоду має місце лише за умов, що вирази в дужках попарно взаємно прості та, обов'язково мають місце попарні нерівності:

$$m \neq n,\ m \neq l,\ n \neq l. \quad (3.3)$$

Виходячи з ймовірності виникнення небезпечної ситуації з перекриттям шифру величина періоду перетворення не може бути менше ніж $T \geq 2^{64}$, тому з (3.2) отримуємо ще одну оцінку сумарної довжини регістрів в байтах:

$$m + n + l \geq 64. \quad (3.4)$$

На підставі (3.3) і (3.4), а також вимог щодо примітивності характеристичного полінома отримуємо варіант реалізації довжин регістрів в байтах. Зокрема, припустимим є варіант:

$$m + n + l = 19 + 22 + 23 = 64 \text{ (байти)}. \quad (3.5)$$

Зазначимо, що для початкового заповнення всіх трьох регістрів необхідно більше даних ніж обрана довжина ключу, яка складає загалом 16 байтів. Для узгодження цих потреб сформуємо ключовий розклад.

Позначимо: $K, \overline{K}, \widetilde{K}$ – відповідно початковий ключ, далі ключ, отриманий з початкового шляхом його циклічного зсуву на 1 байт, потім ключ, що отриманий з початкового шляхом його циклічного зсуву на 3 байти.

Під час створення початкового заповнення K завантажується в перший регістр, $\overline{K}$ завантажується в другій, а $\widetilde{K}$ – в третій регістр. Після цього не заповненими залишаються 3 байти першого регістра, 6 байт другого регістра і 7 байтів третього регістра – загалом 16 байтів, які утворюють параметр – синхромаркер S, якій використовується для перезапуску пристрою після випадкового збою.

Для забезпечення якісного початкового заповнення до початку шифрування алгоритм має відпрацювати так, щоб найдовший регістр оновився що найменш двічі. Для надійного вирівнювання пропонується забезпечити 128 тактів початкового прогону. Початковий прогін відбувається без управління рухом регістрів.

Точки зворотного зв'язку та схеми управління рухом для всіх регістрів залишаються незмінними (як в A5/1), що забезпечує необхідні якості характеристичних поліномів та управління рухом.

Відмінність полягає лише у використанні бітів в байтах управління рухом, а саме, порівняння здійснюється лише перших бітів відповідних байтів.

В процесі робочого режиму схема формує у кожному циклі роботи один байт шифруючої послідовності.

Зауважимо, що максимальний період кожної з лінійних рекурентних послідовностей в обраній схемі в кільці лишків $Z/_{2^\mu}$згідно [28] оцінюється як:

$$T(R1) \leq (2^m - 1) \cdot 2^{\mu-1},$$
$$T(R2) \leq (2^n - 1) \cdot 2^{\mu-1},$$
$$T(R3) \leq (2^l - 1) \cdot 2^{\mu-1},$$

де m, n, l – степені відповідних характеристичних многочленів.

А це означає, що у випадку вибору многочленів з урахуванням відповідних вимог [28] загальний період вихідної послідовності, якій є найменшим спільним кратним величин $T(R1),\ T(R2),\ T(R3)$ (без урахування схеми керування рухом регістрів), взагалі кажучи, збільшується. В свою чергу, це свідчить про покращення криптографічних якостей модифікованої схеми.

Наступний елемент вдосконалення криптосхеми – вузол накладання шифру, який реалізований за допомогою бітової операції XOR.

В [101, 104] доведено, що з точки зору забезпечення конфіденційності та імітостікості інформації найбільш ефективним вузлом накладання шифру є вузол багатоалфавітної заміни.

Формування випадкової багатоалфавітної заміни потребує суттєвих обчислювальних ресурсів, тому для модернізації вузла пропонується схема «латинський квадрат» [78, 57]. Згідно з теоремою Шеннона латинські квадрати є основою ідеальних шифрів: «Ідеальні системи, в яких кількість криптограм, кількість повідомлень і кількість ключів однакові, характеризуються такими властивостями, що 1) кожне M з'єднане з кожним E рівно однією лінією, 2) усі ключі однакові, ймовірно. Таким чином, матричним зображенням системи є латинський квадрат» [78, с. 68].

В загальному випадку латинський квадрат є таблицею, в якій перший рядок є перестановкою символів, що підлягають зашифруванню, а кожний наступний рядок є результатом циклічного зсуву попереднього рядка на один символ. А саме, якщо X є підстановкою заміни степеня n, тобто

$$X = \begin{pmatrix} 1 & 2 & \dots & n \\ x_1 & x_2 & \dots & x_n \end{pmatrix}, \tag{3.6}$$

тоді латинський квадрат, що породжується цією підстановкою має вигляд:

$$L(X) = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ x_2 & x_3 & \dots & x_1 \\ \dots & \dots & \dots & \\ x_n & x_1 & \dots & x_{n-1} \end{pmatrix}. \tag{3.7}$$

Рівняння зашифрування за допомогою латинського квадрату $L(X)$ має наступний вигляд:

$$C_i = L(X, M_i, G_i), \tag{3.8}$$

де $M_i, G_i, C_i \in \{0,1,\dots,255\}$ – відповідно: черговий байт відкритих даних, поточне значення байту шифруючої послідовності та байт зашифрованих даних.

Суть перетворення (3.8) полягає у виборі в якості зашифрованого байту значення в таблиці (3.7) що знаходиться на перетині рядка з номером $G_i + 1$ та стовпчика $M_i + 1$.

Для нашого випадку – побайтного шифрування даних – таблиця латинського квадрату $L(X)$ матиме розмір $2^8 \times 2^8 = 65536$ (64 кбайт), що може бути дуже великим значенням для обраного ПООР.

Тому з метою уникнення необхідності збереження в пам'яті пристрою рівняння зашифрування чергового байту M_i за допомогою поточного значення G_i байту шифруючої послідовності з використанням латинського квадрату (3.8) можна записати у вигляді виразу:

$$C_i = X(M_i + G_i mod\ 2^8). \qquad (3.9)$$

При цьому для збереження підстановки X потрібно лише 256 байт запам'ятовуючого пристрою.

Звернемо увагу, що використання унікальної для кожного пристрою підстановки X забезпечуватиме безпеку інших пристроїв в разі компрометації цього елементу для будь якого окремого пристрою [111].

Враховуючи запропоновані зміни, модифікована криптосхема алгоритму матиме вигляд як на рис. 3.4 та має назву A5-128.

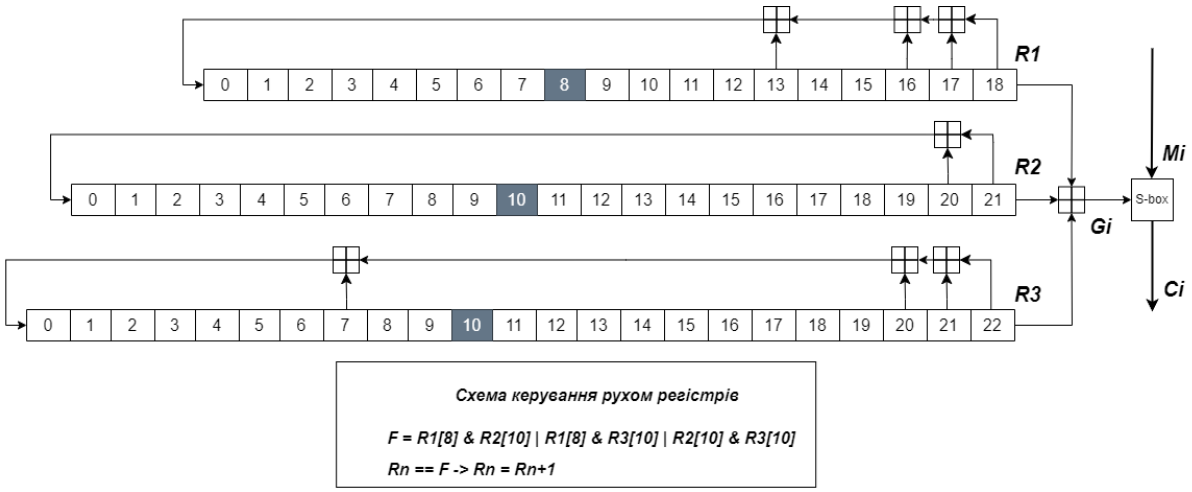

Рис. 3.4. Криптосхема модифікованого алгоритму A5-128

Підсумовуючи викладене, запропоновані зміни криптосхеми в частині заміни бітової обробки даних на байтову не тільки сприяють покращенню криптографічних якостей порівняно з прототипом, а й підвищують зручність застосування модифікованого алгоритму на ПООР.

3.2. Особливості реалізації криптоалгоритму A5-128

Імплементація криптоалгоритму A5-128 передбачає реалізацію окремих його складових з урахуванням обмежень ПООР класу C0 для забезпечення високої швидкості шифрування. Для шифрування інформації за допомогою A5-128, без урахування етапів обміну ключами та синхронізації параметрів, можна виділити такі складові які потребують особливої уваги:

- генерація ключа та синхромаркера: мають бути випадковими та рівномірно розподіленими;
- операції зсуву та керування рухом регістрів мають бути оптимізовані для виконання на ПООР класу C0 з урахуванням, що кожна чарунка регістру в модифікованому алгоритмі має розмір 8 біт;
- формування підстановки заміни X: підстановка заміни повинна мати випадкове та рівномірне розподілення без повторів, алгоритм формування підстановки заміни повинен мати високу швидкодію.

Розглянемо етап генерації криптографічних параметрів ключа K, та синхромаркера S.

Генератори справжніх випадкових чисел використовують недетерміновані джерела для створення випадкових чисел. Один із способів – використовувати фізичний процес, який неможливо передбачити. Джерелами ентропії в даному випадку можуть бути випадкові явища у природі, такі як: тепловий шум, видача випадкових електронів напівпровідником, або фонове випромінювання лічильника Гейгера, але це є дорогим та складним варіантом для застосування у промислових масштабах.

Для формування ключа та синхромаркера на ПООР можна використовувати шум з непідключених контактів аналогово-цифрового перетворювача (АЦП).

У роботі [48] автори досліджують статистичні властивості даних отриманих від непідключеного АЦП на мікроконтролері Arduino Duemilanove. Автори отримали результати які свідчать про те, що молодші біти зчитаних значень демонструють випадкову поведінку, та в режимі зчитування двох молодших біт проходять статистичні тести які не були пройдені в інших режимах збору випадкових значень, на більшості комп'ютерів до якого підключено мікроконтролер. Така поведінка може свідчити про те, що шуми в навколишньому середовищі є невеликими, тому їх вплив можна побачити тільки на молодших бітах зчитаних значень. Вбудовані АЦП є на більшості мікроконтролерів, а у випадку відсутності такого модуля, він може бути включений додатково. Основною перевагою є низька ціна таких модулів близько 1.5 доларів та сумісність з практично усіма мікроконтролерами, в порівнянні, наприклад, з лічильниками Гейгера ціна яких близько 300 доларів.

Таким чином при використанні такого джерела ентропії необхідно враховувати лише молодші біти зчитаного значення, оскільки вони демонструють більш випадкову поведінку [48]. При цьому додавати поточне значення молодшого біта $n[i]$ тільки якщо $n[i] ! = n[i-1]$.

Окрім цього під час формування ключа та синхромаркера необхідно передбачити перевірку згенерованого значення за допомогою частотного монобітного тесту по [76, с. 24], задля протидії похибкам системи та потенційного зовнішнього впливу на контролер:

$$P = erfc(\frac{S_{abs}}{\sqrt{2}}), \tag{3.10}$$

де S_{abs} – абсолютна величина суми X_i по всій довжині послідовності, поділена на корінь квадратний з довжини. Тут $X_i = 2 * \varepsilon_i - 1$, $X_i \in \{-1,1\}$;
$erfc$ – комплементарна функція похибки.

Якщо отримане значення $P < 0.01$, послідовність необхідно відхилити та повторити спробу, після певної кількості невдач, необхідно сповістити шлюз про помилку в роботі системи та неможливості сформулювати ключ. Блок-схема генерації ключа зображена на рис. 3.5. На блок-схемі продемонстровано процес генерації ключа, синхромаркер S пропонується генерувати аналогічно. Варто зауважити, що синхромаркер в даному випадку не є секретним параметром, та дає змогу відновити втрачене з'єднання, шляхом початкового прогону регістрів на основі ключа та нового значення синхромаркера.

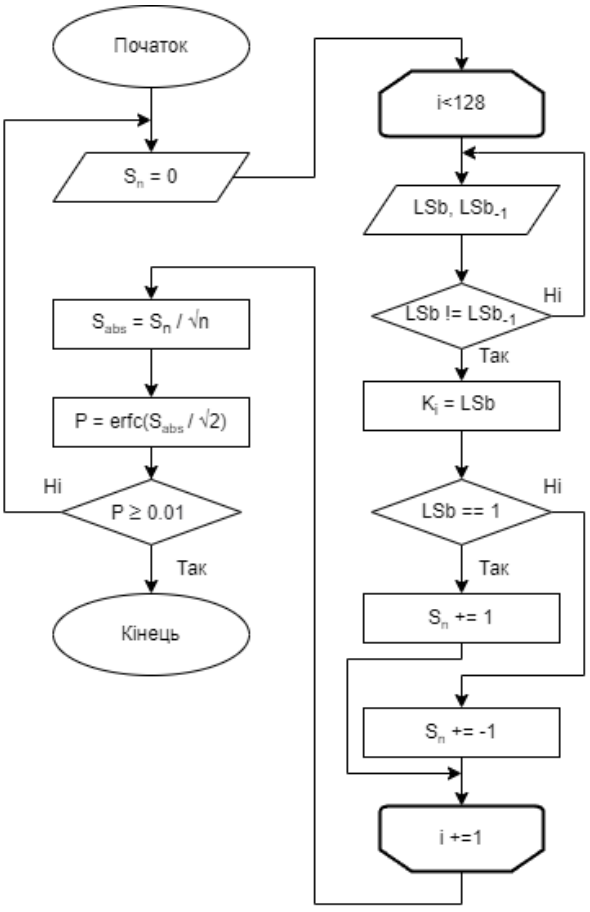

Рис. 3.5. Блок-схема генерації ключа

Для взяття конкретних біт з числа можна скористатись операцією побітового І, об'єднавши дані з бітовою маскою. Для вилучення молодшого біта маска матиме вигляд: 0b00000001. Самі ж біти можна додати до змінної за допомогою побітового АБО.

Варто зауважити, що генерування криптографічних параметрів процес доволі рідкісний, тому не матиме великого впливу на швидкість шифрування.

Розглянемо реалізацію зсуву регістрів та керування зсувом. Визначивши кожну чарунку регістру як 8 бітне значення, постає питання як в програмній реалізації ефективно з точки зору мінімізації тактів процесору, виконувати операцію зсуву з такою структурою. Можна припустити декілька підходів визначення такого регістра в програмній реалізації, найочевидніший з яких представлення регістру у вигляді певної структури даних: масиву, циклічного списку, черги. З огляду на те, що нове значення регістру формується з байтів зворотного зв'язку, необхідно мати вказівники на ці чарунки. Таким чином можна змінювати вказівники на чарунки, замість модифікації всього регістру, таке рішення дозволить досягти максимальної швидкодії при програмній реалізації. Схематичне представлення такого рішення представлене на рис. 3.6, наприклад нехай дано регістр довжиною $l = 5$, з точками зворотного зв'язку x^5, x^4, x^2. Тоді з кожним наступним кроком необхідно зменшувати номер відповідної точки зв'язку, а результат буде записуватись замість останнього значення. Очевидно, зміна номеру повинна відбуватись в межах l.

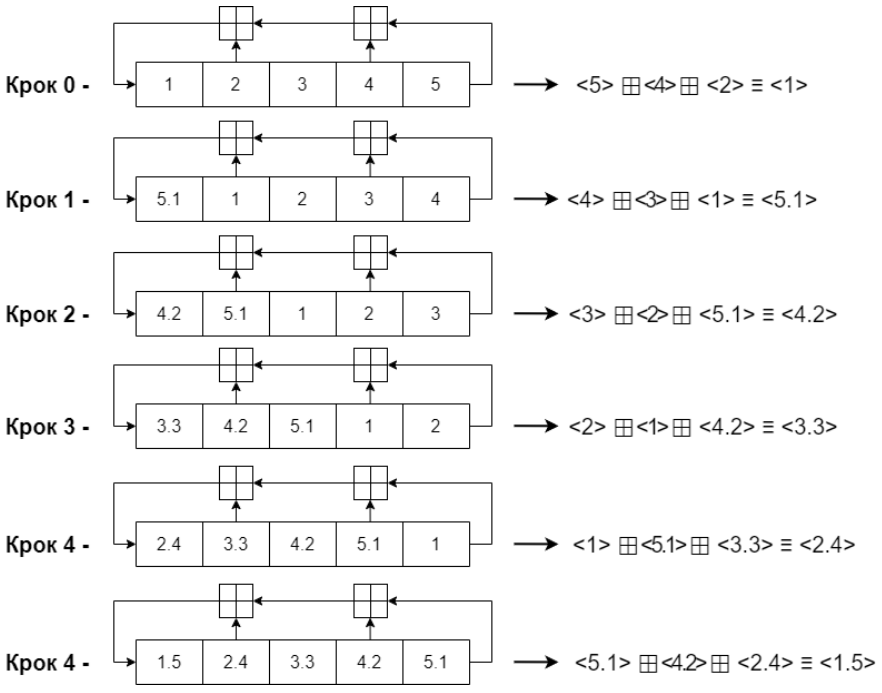

Рис. 3.6. Схема керування рухом регістру

Програмна реалізація такого алгоритму для мінімізації кількості обчислюваних ресурсів для виконання операцій з регістрами зображена на лістингу 1, на прикладі регістру R3. Основна ідея такої реалізації полягає у відслідковуванні необхідних байтів регістру для формування нового значення та керування зсувом. Після кожного «зсуву» номер комірки зменшується на 1, необхідно тільки відслідковувати, щоб номер лежав у допустимому для регістра діапазоні, операція mod в даному випадку не використовується через те, що вона потребує більше тактів, аніж відслідковування переповнення змінної за допомогою умовного оператора. Необхідно зауважити, що номери точок для управління рухом регістрів також змінюються відповідно на -1.

Лістинг 1

```
void shiftReg3(void) {
   reg3[reg3_22] = (reg3[reg3_22] + reg3[reg3_21]+reg3[reg3_20]+reg3[reg3_7])%256;
   if (--reg3_22 == 255) reg3_22 = 22;
   if (--reg3_21 == 255) reg3_21 = 22;
   if (--reg3_20 == 255) reg3_20 = 22;
   if (--reg3_7 == 255) reg3_7 = 22;
   if (--reg3Clock == 255) reg3Clock = 22;
}
```

Як було зазначено схема управління рухом для всіх регістрів залишається такою ж як і в оригінальному алгоритмі. Проте різниця полягає в тому, що в оригінальному алгоритмі значення чарунку було бітовим, а в модифікованому байтове. Зважаючи на таке прийнято рішення використовувати молодший біт відповідного байту для керування рухом (рис. 3.7).

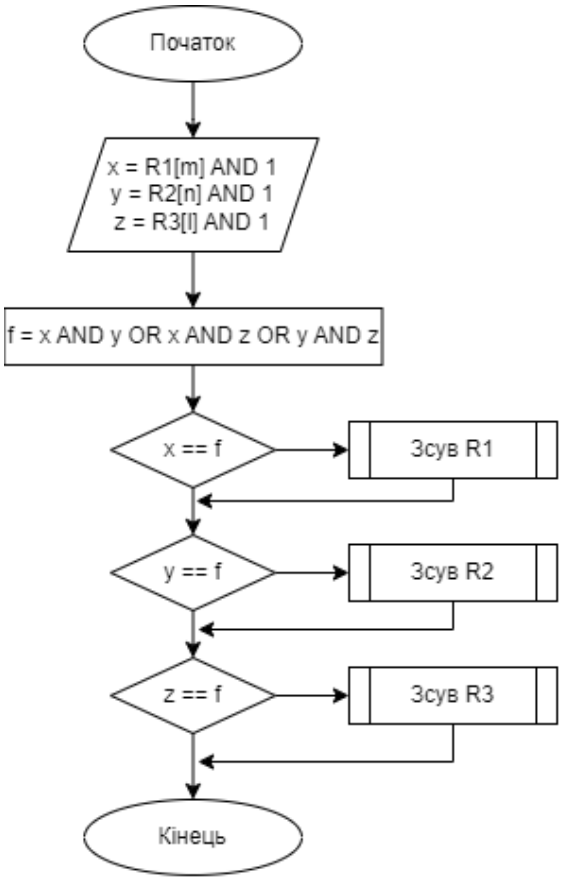

Рис. 3.7. Блок-схема роботи керування рухом регістрів A5-128

Відповідно байт шифруючої послідовності G_i буде формуватись з байтів регістрів зсуву за модулем 256: $G_i = (R1[19] + R2[22] + R3[23])\ mod\ 2^8$. Лістинг реалізації регістрів, управління рухом та формування шифруючої послідовності в модифікованому алгоритмі представленні на додатку Б.

Розглянемо процес формування підстановки заміни. З урахуванням обчислювальних ресурсів ПООР, зберігати весь латинський квадрат недоцільно, в свою чергу збереження підстановки X потребує лише 256 байт пам’яті. Постає питання яким чином формувати підстановку заміни з високою швидкістю та використанням мінімальної кількості обчислювальних ресурсів.

Як визначено у [101] базові методи сучасної криптології, що забезпечують перетворення псевдовипадкових послідовностей у послідовність підстановок заміни відповідного степеню мають деякі недоліки. Вони полягають у складності оперативної заміни системи базових

підстановок, складність виявлення можливих збоїв та помилок, довгий процес формування підстановок.

Зважаючи на таке для формування підстановки заміни X вирішено використовувати швидкий алгоритм генерації підстановок багатоалфавітної заміни Складанного [101].

Згідно алгоритму Складанного для генерації підстановки степеню n: $X = \begin{pmatrix} 0 & \cdots & n-1 \\ x_0 & \cdots & x_n - 1 \end{pmatrix}$ за допомогою послідовності випадкових чисел $i_0, i_1, \dots, i_{n-1} \in Z_n$, необхідно:

1) Визначити черговий перехід: $x_k = i_m$

2) Додати перехід до множини сформованих переходів: $A = A \cup \{x_k\}$

3) Обчислити номер наступного переходу: $k = k + 1 \; mod \; n$

4) Знайти номер чергового випадкового числа $m = m + 1$, якщо $m = n$, перейти в кінець

5) Якщо $i_m \in A$ перейти до визначення наступного переходу, інакше

6) Виконати модифікацію $i_m = i_m + \delta \; mod \; n$, та перейти до перевірки визначених переходів

7) Останньому переходу призначити $x_k = Z_n / A$

Перевагою цього алгоритму є те, що він генерує рівномірно розподілену послідовність за фіксовану кількість кроків, та є більш швидким порівняно з методом безповторного набору, що є дуже важливим при реалізації на ПООР.

Останнім етапом реалізації алгоритму є метод шифрування повідомлення. В модифікованому алгоритмі вузол накладання шифру реалізований за допомогою шифру багатоалфавітної заміни. Як було визначено, для зберігання таблиці латинського квадрата необхідно 64 кБайт пам’яті, відповідно на ПООР класу С0, це більше половини від всього обсягу пам’яті. В той час коли на обраній платформі для реалізації всього 32 кБайт ПЗП тому з метою економії ПЗП використовується вираз (3.9).

Наприклад, нехай підстановка заміни $X = \begin{pmatrix} 0\ 1\ 2\ 3\ 4 \\ 3\ 2\ 1\ 4\ 0 \end{pmatrix}$, шифруюча послідовність - $G_i \in \{0,1,2,3,4\}$, повідомлення, що підлягає зашифруванню - $M_i \in \{0,1,2,3,4\}$.

Якщо для поточного символу повідомлення $G_i = 3$, $M_i = 3$, тоді $\mathrm{C}_i = (4+4)\ mod\ 5 = 1 = X(1) = 2$.

Латинський квадрат у цьому випадку має вигляд:

$$L(X) = \begin{pmatrix} 3\ 2\ 1\ 4\ 0 \\ 2\ 1\ 4\ 0\ 3 \\ 1\ 4\ 0\ 3\ 2 \\ 4\ 0\ 3\ \mathbf{2}\ 1 \\ 0\ 3\ 2\ 1\ 4 \end{pmatrix}$$

В модифікованому алгоритмі довжина підстановки заміни 256 байт, додавання байту шифруючої послідовності та байту даних здійснюється за модулем 256 (рис. 3.8).

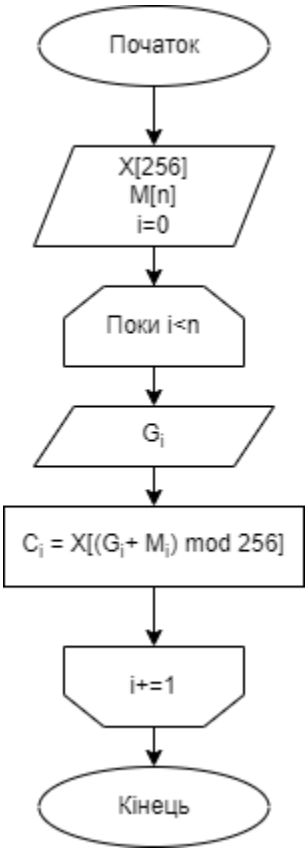

Рис. 3.8. Блок-схема функції вузла накладання шифру

Для розшифрування необхідно в рядку G_i, знайти символ C_i ,відповідний йому стовпчик буде символом вихідного повідомлення M_i.

Варто зауважити, що використання байтової структури в даному випадку дозволяє досягти більшої швидкості, за рахунок того, що операція mod у випадку коли дільник n є степенню двійки замінюється на операцію бітовою операцією AND з $n-1$. Така операція виконується близько в 16 разів швидше за операцію взяття залишку від ділення.

3.3. Оцінка ймовірнісно-статистичних якостей шифруючої послідовності

Враховуючи модифікації алгоритму, необхідно оцінити ймовірнісно-статистичні якості шифруючої послідовності. Випадковість відіграє важливу роль у багатьох областях криптографії. Генерування випадкових чисел є складним завданням, так само як і оцінка якості згенерованих даних. На практиці оцінка випадковості значною мірою покладається на емпіричні тести випадковості [32].

Зважаючи на таке було вирішено провести оцінку розробленого алгоритму в частині генерації випадкових послідовностей. Насправді питання «Що можна вважати випадковим» є настільки філософським наскільки і математичним. Що якщо монета падає 5 разів з 5 аверсом догори, чи означає це що цей метод не є випадковим, або коли по черзі випадає кожна сторона? Неможливо стверджувати напевне, але для цього існують статистичні тести, які на основі великої вибірки даних, можуть виявляти деякі закономірності які можуть свідчити про те, що послідовність невипадкова або ж випадкова.

Більшість емпіричних тестів випадковості засновані на перевірці статистичних гіпотез. Кожен тест порівнює певні характеристики даних (частота одиниць, частота m-бітових блоків тощо) з очікуваною тестовою

статистикою (0.5, 2-m і т.д.), яка попередньо обчислюється для випадкових нескінченних послідовностей. У цьому контексті випадковість є імовірнісною властивістю і може бути охарактеризована та описана в термінах ймовірності. Це пов'язано з тим, що навіть хороший генератор випадкових чисел видає послідовності (наприклад послідовність всіх одиниць) з характеристиками, що значно відрізняються від значень очікуваних у тестах. Тому не можливо відрізнити, чи дана послідовність з "поганими" характеристиками була згенерована дефектним генератором, чи послідовність була випадково згенерована хорошим генератором. В контексті емпіричних тестів випадковості, вона описується як ймовірність того, що ідеальний генератор випадкових чисел згенерує послідовності з такою самою або меншою якістю випадковості, ніж та, що була продемонстрована в аналізованій послідовності.

Для тестування послідовності як було визначено в розділі 2, буде використовуватись набір статистичних тестів NIST STS[76].

За цим набором тестів послідовність вважається випадковою, якщо вона пройшла всі випробування, але навіть дійсно випадкова послідовність демонструє високу ймовірність (близько 80%) того, що вона не пройде принаймні одне випробування NIST STS. Якщо дані не проходять деякі тести, NIST STS рекомендує аналізувати різні вибірки.

Результати статистичних тестів випадковості зазвичай представлені у вигляді значення p, яке представляє ймовірність того, що ідеальний генератор випадкових чисел створить менш випадкову послідовність, ніж послідовність, яка тестується.

Хоча значення p для єдиного тесту на випадковість має чітку статистичну інтерпретацію, інтерпретація результатів наборів тестів є більш складною. Це відбувається тому, що емпіричні тести на випадковість і їхні результати зазвичай залежать один від одного і корелюються.

Наприклад, якщо частоти одиниць і нулів упереджені (нерівні) для заданої послідовності, ймовірно, що частоти блоків з двох бітів теж упереджені. Щоб надати чітку статистичну інтерпретацію результатів набору

тестів, потрібно проаналізувати залежність/кореляцію між результатами тестів, які застосовуються до випадкових даних.

Набір STS складається з 15 окремих статистичних тестів, кожен такий тест здійснює перевірку бінарної послідовності на ознаки відхилення від випадковості. Кожен тест сформульований для оцінки нульової гіпотези, яка полягає в тому, що послідовність, яка тестується, є випадковою. Статистична ознака тесту є функцією протестованих даних, яка стискає виміряну якість випадковості в єдине значення, спостережувану статистичну ознаку.

Щоб оцінити тест, розподіл статистичної ознаки має бути відомим за нульовою гіпотезою (коли очікується, що дані будуть випадковими). Більшість тестів STS NIST мають розподіл x^2 або нормальний як свій референтний розподіл. Спостережувана статистична ознака зазвичай перетворюється на p-значення за допомогою референтного розподілу, оскільки p-значення можна легше інтерпретувати. Це значення представляє ймовірність того, що ідеальний генератор випадкових чисел створить послідовність менш випадковою, ніж послідовність, що аналізується [119].

Найважливішою властивістю p-значення є те, що для довільних статистичних тестів (і не лише для тестів на випадковість), які задовольняють нульову гіпотезу, значення p рівномірно розподілені на інтервалі (0, 1). Це означає, що випадкові послідовності, оброблені довільним емпіричним тестом, повинні рівномірно розподілятися на (0, 1). Тому ймовірність того, що p, обчислені для випадкової послідовності, лежать в інтервалі [a, b], можна висловити як: $P(a \leq p \leq b) = b - a$.

Рівень значущості α є імовірністю того, що послідовність буде сприйнята як невипадкова. Якщо отримане значення більше або дорівнює α, це означає що текст пройдено, і гіпотеза про те, що послідовність є випадковою може бути підтверджена. Рівень значущості рекомендований NIST $a = 0.01$ [76]. Такий рівень значущості і був відібраний для тестування послідовності.

Кожен тест STS NIST визначається статистичним показником одного з наступних трьох типів і досліджує випадковість послідовності відповідно до:

1. Бітів – ці тести аналізують різні характеристики бітів, такі як пропорція бітів, частота зміни бітів і накопичувальні суми;
2. Блоки m-біт – ці тести аналізують розподіл m-бітних блоків (m зазвичай менше 30 біт) у послідовності або її частинах;
3. Групи M-біт – ці тести аналізують складні властивості M-бітних (M зазвичай більше 1000 біт) частин послідовності, такі як ранг послідовності, побачений як матриця, спектр послідовності або лінійна складність потоку бітів.

Всі тести параметризовані n, що позначає довжину двійкової послідовності, яка підлягає тестуванню. Деякі тести також параметризовані другим параметром, позначеним m або M. Оскільки еталонні розподіли статистичних ознак тестів STS NIST наближені асимптотичними розподілами, то тести дають точні результати (p-значущі) лише для певних значень їх параметрів. Значення параметрів для кожного конкретного тесту брались відповідно до рекомендацій NIST [76].

Деякі з тестів STS NIST виконуються в декількох варіантах, тобто вони виконують декілька підтестів і досліджують більше властивостей послідовності того ж типу. Наприклад, тест накопичувальної суми досліджує послідовність відповідно вперед і назад накопичувальної суми.

Щоб застосувати всі тести, параметр n (довжина біт послідовностей) повинен бути більше 100000. Документація NIST STS рекомендує, щоб принаймні $k = \alpha - 1 = 100$ послідовностей були протестовані. Це також є відповідним значенням для тесту однорідності p-значень (принаймні 55 послідовностей повинні бути оброблені). Оскільки p-значення обробляються NIST STS за допомогою деякої апроксимації, чим більше послідовностей буде протестовано, тим більш точні результати будуть отримані.

Проте варто зауважити, що деякі тести для отримання коректного результату вимагають мінімум $n = 10^6$ бітів для тестування.

Для формування початкового заповнення регістрів модифікованого алгоритму відповідно до розробленої реалізації було обрано популярний мікроконтролер ATmega328 з такими характеристиками:

– тактова частота – 16 МГц;
– ПЗП-пам’ять – 32 кБ;
– ОЗП – 2 кБ;
– EEPROM – 1 кБ;

У відповідність до класифікації цей пристрій відноситься до класу C0, що цілком відповідає кінцевій платформі для якої розробляється метод.

Проте мікроконтролер використовувався лише для формування початкового заповнення, послідовність генерувалась на стаціонарному комп’ютері, оскільки формування вибірки достатньої для тестів зайняло б дуже багато часу якби вона повністю формувалась на мікроконтролері. Проте самі значення, що генеруються є однаковими на будь-якому пристрої за умови однакових точок зворотного зв’язку та схеми управління рухом.

Для проходження тестів було згенеровано m = 1000 послідовностей біт по n = 10^6 кожна, тобто загальний розмір набору даних для тесту $n \geq 10^9$. Початкове значення змінювалось після досягнення періоду. Модифікований алгоритм повертає один байт шифруючої послідовності на кожному такті алгоритму, всі вісім біт записувались до тестової послідовності.

До всіх послідовностей та тестів був застосований рівень значущості $a = 0{,}01$, а всі інші параметри необхідні для кожного тесту вказані в табл. 3.1.

За результатами статистичних досліджень шифруючих послідовностей $\{G_i\}$ на основі [105] з використанням рекомендацій [76] були отримані наступні результати.

В [76] пропонується дві стратегії ухвалення рішення, щодо проходження тестів на випадковість.

Згідно *першої стратегії* необхідно визначити частку послідовностей $P1$, які пройшли перевірку, тобто $p > a$, та порівняти її з нижньою межею довірчого інтервалу $P1_{THR}$, результати представлено в табл. 3.1.

$P1 = \frac{\sum_{i=1}^{1000}(P\ value(i) \geq a)}{m}$, $P1_{THR} = (1-a) \pm 3\sqrt{\frac{(1-a)a}{m}} = 0.9805607$

Якщо для одного з 15 тестів значення $P1$ виходить за ці межі, вважається, що тест не пройдено.

Таблиця 3.1

Результати тестування шифруючої послідовності згідно першої стратегії

Тест №	Назва тесту і його параметри	P1
1	**Частотний монобітний**	0.993
2	**Частотний блочний** (M=128)	0.996
3	**Серій**	0. 990
4	**Довгих серій одиниць** (M=10000)	0.992
5	**Рангу випадкової {0,1}-матриці**	0.989
6	**Дискретного перетворення Фур'є**	0.989
7…154	**Відповідності аперіодичних шаблонів, що не перекриваються** (M=9, 148 шаблонів)	0.990 (середнє)
155	**Відповідності періодичних шаблонів, що перекриваються** (M=9)	0.988
156	**Лінійної складності** (M=500)	0.984
157	**Універсальний статистичний – Маурера** (L=8, Q=2356)	0.987
158	**Послідовності** (M=16, $\nabla\psi^2{}_m\ (obs)$)	0.996
158	**Послідовності** (M=16, $\nabla^2\psi^2{}_m\ (obs)$)	0.986
160	**Наближеної ентропії** (M=10)	0.996
161	**Накопичених сум** (Прямий)	0.994
162	**Накопичених сум** (Зворотній)	0.991
163…170	**Випадкових відхилень** (x = –4,…,–1, 1,…,4)	0.989
171…188	**Вигляду випадкових відхилень** (x = –9,…,–1, 1,…,9)	0.989

Згідно *другої стратегії* розподіл P для кожного тесту повинен бути рівномірним на інтервалі [0,1].

$$x^2 = \sum_{i=1}^{10} \frac{(Ci - m/10)^2}{m/10},\ P2 = P(x^2) = igamc(\frac{9}{2}, x^2/2).$$

Якщо отримане значення в результаті тестування $P1 < 0.0001$, то вважається, що ГПВЧ тест не пройшов.

Для перевірки цієї стратегії значення P були розбиті на 10 підінтервалів C1-C1, з кроком 0.1. Результати наведені у табл. 3.2.

Таблиця 3.2

Результати тестування шифруючої послідовності згідно другої стратегії

Тест №	Назва тесту і його параметри	P2
1	**Частотний монобітний**	0.624627
2	**Частотний блочний** (M=128)	0.065230
3	**Серій**	0.518106
4	**Довгих серій одиниць** (M=10000)	0.117432
5	**Рангу випадкової {0,1}-матриці**	0.205531
6	**Дискретного перетворення Фур'є**	0.370262
7…154	**Відповідності аперіодичних шаблонів, що не перекриваються** (M=9, 148	0.510992
155	**Відповідності періодичних шаблонів, що перекриваються** (M=9)	0.783019
156	**Лінійної складності** (M=500)	0.866097
157	**Універсальний статистичний – Маурера** (L=8, Q=2356)	0.574903
158	**Послідовності** (M=16, $\nabla\psi^2{}_m\ (obs)$)	0.457825
158	**Послідовності** (M=16, $\nabla^2\psi^2{}_m\ (obs)$)	0.984415
160	**Наближеної ентропії** (M=10)	0.893482
161	**Накопичених сум** (Прямий)	0.585209
162	**Накопичених сум** (Зворотній)	0.274341
163…170	**Випадкових відхилень** (x = –4,…,–1, 1,…,4)	0.426403
171…188	**Вигляду випадкових відхилень** (x = –9,…,–1, 1,…,9)	0.342937

Згідно проведеному тестуванню, модифікована версія алгоритму пройшла всі статистичні тести. Таким чином підтверджена нульова гіпотеза, і

можна стверджувати, що згенерована шифруюча послідовність може вважатись випадковою рівномірно розподіленою.

Отже, можна стверджувати, що запропонований модифікований алгоритм за умов випадкової генерації ключа, може застосовуватись для забезпечення конфіденційності та імітостікості повідомлень, що обробляються пристроями з обмеженими обчислювальними ресурсами.

3.4. Оцінка рівня інформаційного ризику з використанням алгоритму A5-128 в незахищених протоколах

Оцінка ризиків є одним з фундаментальних компонентів організаційного процесу управління ризиками [43]. Оцінки ризиків використовуються для виявлення, оцінки та пріоритизації ризиків для організаційних операцій (тобто місії, функцій, іміджу та репутації), організаційних активів, осіб, інших організацій, що виникають в результаті експлуатації та використання інформаційних систем.

Мета оцінок ризиків – проінформувати людей, що приймають рішення та надати ризикові відповіді, визначивши:

- пріоритетні загрози для організацій або загрози, спрямовані через організації проти інших організацій;
- вразливості як внутрішні, так і зовнішні для організацій;
- вплив (тобто шкода) для організацій, яка може виникнути з урахуванням потенціалу для загроз, що експлуатують вразливості;
- імовірність того, що шкода виникне.
- кінцевим результатом є визначення ризику (тобто, як правило, функція ступеня шкоди та ймовірності виникнення шкоди).

Оцінки ризиків можуть проводитися на всіх трьох рівнях у ієрархії управління ризиками, включаючи рівень 1 (рівень організації), рівень 2 (рівень

місії/бізнесу) та рівень 3 (рівень інформаційної системи). На рівнях 1 і 2 організації використовують оцінки ризиків для оцінки, наприклад, системних інформаційних ризиків, пов'язаних з організаційним управлінням та управлінськими діяльністю, місії/бізнес-процесами, архітектурою підприємства або фінансуванням програм інформаційної безпеки. На рівні 3 організації використовують оцінки ризиків для більш ефективної підтримки впровадження. Рамки управління ризиками (тобто, класифікація безпеки; вибір, впровадження та оцінка контрольних заходів безпеки; авторизація інформаційної системи та загального контрольного; та моніторинг контрольних заходів безпеки). В даному розділі буде проводитись оцінка поточного рівня інформаційного ризику саме розробленого методу, тобто рівень 3.

Моделі ризику визначають фактори ризику, які будуть оцінені та взаємозв'язки між цими факторами.

Фактори ризику – це характеристики, що використовуються в ризикових моделях як вхідні дані для визначення рівня ризику в оцінці ризику. Типові фактори ризику включають загрози, вразливості, вплив, імовірність і схильність. Фактори ризику можуть бути декомпозовані на більш детальні характеристики (наприклад, загрози декомпозовані на джерела загроз і події загрози). Ці визначення важливі для організацій для документування до проведення оцінки ризиків, оскільки оцінки ґрунтуються на добре визначених атрибутах загроз, вразливостей, впливу та інших факторів для ефективного визначення ризику.

Загроза – це будь-яка обставина або подія, яка може негативно вплинути на операції організації та активи, індивідів, інших організацій або Націю через інформаційну систему шляхом несанкціонованого доступу, знищення, розкриття або модифікації інформації та / або відмови в обслуговуванні. Події загрози викликаються джерелами загрози. Джерело загрози характеризується як:

– намір і метод, спрямований на експлуатацію вразливості;

– або ситуація і метод, які можуть випадково експлуатувати вразливість.

У загальному випадку типи джерел загроз включають:

– ворожі кібернетичні або фізичні атаки;
– людські помилки діяльність або бездіяльність;
– структурні несправності контрольованих організацією ресурсів (наприклад, апаратне забезпечення, програмне забезпечення);
– природні та техногенні катастрофи, аварії та несправності, що не підконтрольні організації.

При оцінці загроз використовувались методичні рекомендації щодо впровадження системи управління інформаційною безпекою та методики оцінки ризиків відповідно до стандартів Національного банку України [115].

Виходячи з побудованої моделі загроз, можна зробити висновок, що використання лише алгоритму шифрування є недостатнім для нейтралізації всіх визначених загроз.

Однією з загроз які не можна повністю нейтралізувати криптографічними методами є фізичний контур безпеки. Це зумовлено тим, що пристрої IoT зазвичай орієнтовані на модель M2M та можуть встановлюватись в різних місцях в залежності від потреб. Тому цей ризик необхідно оцінювати з огляду на місце розташування контролерів. В деяких випадках їх фізична охорона може бути недоцільною з фінансової точки зору. Тому необхідно оцінювати інші фактори: наскільки критичним буде вихід з ладу одного пристрою, або його компрометація зловмисником, фізичне пошкодження. Оскільки в модифікованому алгоритмі вузол накладання шифру змінений на латинський квадрат з використанням підстановки заміни, компроментація одного пристрою ніяк не вплине на функціонування мережі, через те, що кожен пристрій використовує унікальний ключ шифрування та підстановку заміни. Але також необхідно врахувати можливість підміни з метою імітації недійсної обстановки. Одним із варіантів зменшення впливу від такої загрози може бути зменшення критичних параметрів, що зберігаються на

пристрої. Нажаль, криптографічними методами не можливо нейтралізувати загрозу, що полягає у фізичному знищенні пристрою. Проте система повинна перевіряти стан підключеного обладнання, щоб у випадку виникнення помилки сповістити про це шлюз.

Інша категорія вразливостей пов’язана з використанням відкритих каналів зв’язку. З такої вразливості випливають загрози модифікації, фальсифікації інформації. Для нейтралізації вказаних загроз необхідне введення унікальних ідентифікаторів для пристроїв та безпечне управління ключами. Цього можна досягти за рахунок побудови криптографічного протоколу для забезпечення безпечного формування сеансових ключів, та безпечної їх передачі між ПООР та шлюзом.

З огляду на таке були визначені загрози котрі потребують додаткових механізмів захисту задля їх нейтралізації.

Висновки до третього розділу

1. Розроблено метод криптографічного захисту інформації в мережі IoT на основі модифікованого алгоритму A5/1, що забезпечує підвищену стійкість шифрування та імітостійкість завдяки застосуванню байтової обробки інформації та застосування вузла накладання шифру на основі змінного латинського квадрату.

2. Визначені особливості програмної реалізації модифікованого алгоритму з урахуванням обмежених обчислювальних ресурсів пристроїв класу C0. Розроблено алгоритм формування ключа та синхромаркера з використанням фізичних величин (тобто дійсно випадкових) – аналогового шуму. Розроблено алгоритм реалізації зсуву регістрів з урахуванням розміру кожної чарунки 8 біт, алгоритм забезпечує високу швидкодію виконання зсуву регістра. Для генерації підстановки заміни визначено до використання

швидкий алгоритм генерації підстановок багатоалфавітної заміни Складанного. Розроблені та підібрані алгоритми повинні забезпечувати високу швидкодію на ПООР класу С0.

3. Проведено оцінку ймовірнісно-статистичних якостей шифруючої послідовності алгоритму А5-128. Для оцінки використовувався набір тестів NIST STS. У всіх тестах до згенерованої шифруючої послідовності, отримано результат згідно якого можна підтвердити гіпотезу про те, що послідовність може вважатись випадковою та рівномірно розподіленою.

4. Проведено оцінку поточного рівня інформаційного ризику при використанні незахищених протоколів зв’язку. Проаналізовані найімовірніші вразливості стосовно функціонування методу захисту. На основі аналізу визначено необхідність забезпечення надійної ідентифікації пристроїв та механізму безпечного управління ключами. Таким чином визначена потреба в побудові захищеного протоколу зв’язку в мережі IoT з використанням ПООР.

РОЗДІЛ 4 АНАЛІЗ ЕФЕКТИВНОСТІ МОДИФІКОВАНОГО АЛГОРИТМУ A5-128 ДЛЯ ЗАХИСТУ ІНФОРМАЦІЇ В МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ

4.1. Дослідження платформ для проведення експерименту

Алгоритми КПЗВ NIST призначені для ефективного захисту даних з використанням мінімальної кількості обчислювальних ресурсів. Тому доречно проводити аналіз ефективності модифікованого алгоритму за показниками, що визначені NIST для КПЗВ. А саме:

- розмір ключа;
- розмір блоку тексту;
- використання ПЗП;
- використання ОЗП;
- швидкість шифрування;
- затримка шифрування;
- споживана потужність шифрування за секунду;
- споживана потужність за біт.

Окрім того необхідно також враховувати криптографічні якості алгоритму, в тому числі криптостійкість та імітостійкість шифрування.

Для аналізу модифікованого криптографічного алгоритму A5-128 обрано такі апаратні платформи як: Arduino Uno та Raspberry Pi. Такий вибір зумовлений тим, що як було зазначено раніше деякі алгоритми працюють гірше на пристроях класу C0, а платформа Arduino Uno відноситься до цього класу пристроїв, окрім цього вона є розповсюдженою платформою для мереж IoT. Платформа Raspberry Pi була обрана в якості проміжного шлюзу між пристроєм з обмеженими обчислювальними ресурсами та сервером, оскільки за класифікацією наведеною раніше вона не відноситься до ПООР, проте часто

використовується в мережах IoT де необхідно більше обчислювальної потужності.

Термін Arduino означає відкриту платформу фізичного комп'ютингу, що базується на мікроконтролері AVR і середовищі для написання програм для мікроконтролера. Мікропроцесорна плата використовується для розробки інтерактивних об'єктів шляхом прийняття вхідних даних від різних перемикачів або датчиків і керування різноманітними світловими приладами, актуаторами та іншими фізичними вихідними пристроями [3].

Arduino має три основні види пам'яті:

– постійна енергонезалежна – це простір пам'яті, який використовується для зберігання довгострокової інформації.
– оперативна – це місце, де зберігаються, обробляються та модифікуються змінні.
– тривала енергонезалежна флеш-пам’ять, також відома як «програмний простір», є місцем, де зберігається програма Arduino. Флеш-пам'ять використовується для зберігання програми та будь-яких ініціалізованих даних.

Arduino Uno – це компактна плата з 8-бітним мікроконтролером ATmega328p який працює на частоті 16 МГц, має 32 кБ флеш-пам'яті, 2 кБ ОЗП та 1 кБ EEPROM.

Характеристики цього контролеру підходять для проведення тесту, оскільки він має в 5 разів менше ОЗП пам’яті та в 3 рази менше флеш-пам’яті ніж верхня межа пристроїв класу C0. Тобто якщо метод буде ефективний в реалізації на цій платформі, можна вважати, що він є ефективним для пристроїв класу C0.

Сама ж плата вже має необхідні порти вводу/виводу для підключення різних сенсорів та виконавчих пристроїв, що робить її зручним інструментом для розробників пристроїв IoT. ATmega328 має ряд можливостей для здійснення зв’язку з комп’ютером, або іншими пристроями. В ATmega328 є приймач-передавач UART, який дозволяє реалізувати послідовний зв’язок за

допомогою цифрових виводів 0 (RX) і 1 (TX). Мікроконтролер ATmega16U2 на платі забезпечує зв’язок цього приймача-передавача з USB-портом комп’ютера, і при підключенні до ПК дозволяє Arduino визначатися як віртуальний послідовний порт. Прошивка мікросхеми 16U2 використовує стандартні драйвери USB-COM, тому встановлення зовнішніх драйверів не вимагається. У пакеті програмного забезпечення Arduino входить спеціальна програма, яка дозволяє читати та відправляти на мікроконтролер прості текстові дані.

У мікроконтролері ATmega328 також реалізована підтримка послідовних інтерфейсів I2C і SPI. Наявність таких інтерфейсів дозволяє під’єднувати різні модулі для реалізації різних протоколів зв’язку як Bluetooth, WiFi, GSM, 3G, Ethernet, Shockburst і т.д. Таким чином, теоретично є можливість до використання різноманітних протоколів зв’язку за рахунок використання додаткових модулів. Проте варто зауважити, що такі додаткові модулі часто є набагато потужнішими ніж самі пристрої класу C0, наприклад модуль Bluetooth LE на чипі серії nRF51 має від 128 до 256 кБ флеш пам’яті та 16 кБ ОЗП [90], що є більше ніж верхня межа класу C0.

Raspberry Pi в розроблюваній системі буде використовуватись в якості шлюзу, оскільки пристрої класу C0 є дуже обмеженими сенсорними елементами. Через їх доступні обчислювальні ресурси, вони часто не мають можливості для реалізації прямого доступу до інтернету з використанням стандартизованих протоколів. Пристрої класу C0 братимуть участь у комунікаціях з інтернетом за допомогою більших пристроїв, які діють як проксі, шлюзи або сервери [14].

Raspberry Pi 3 B + має 4х-ядерний 64-бітний SoC Broadcom BCM2837B0 процесор, що працює на частоті 1.4ГГц. Wi-Fi двох-діапазонний стандарту IEEE 802.11ac, та Bluetooth - 4.2 BLE. Гігабітний мережевий адаптер Ethernet забезпечує передачу даних до 300 мБ в секунду. Оперативної пам'яті міститься 1 гБ. Постійна пам’ять визначається встановленою карткою пам’яті розміром мінімум 8 гБ для популярних операційних систем.

Загалом за деякими класифікаціями Raspberry Pi також можна віднести до обмежених пристроїв, проте не за класифікацією IETF, в якій чітко визначено, що подібні пристрої хоч і можуть бути обмежені з огляду на електроживлення, та вони мають достатньо обчислювальних ресурсів для впровадження в сучасні системи з використанням стандартних протоколів передачі даних в мережах в тому числі полегшених протоколів для IoT, наприклад використання протоколу MQTT в роботі [98].

Рішення обрати Raspberry Pi 3 Model B+ в ролі шлюзу обумовлене достатніми обчислювальними ресурсами та наявністю сучасних інтерфейсів для підключення до існуючих стандартних мереж. Проте розглядались і схожі пристрої за характеристиками, такі як: BeagleBone Black Wireless, ODROID-N2, LattePanda.

4.2. Побудова захищеного протоколу для забезпечення безпеки даних в мережі інтернету речей

Виходячи з побудованої моделі загроз для забезпечення безпеки даних в мережі IoT можливо зробити висновок, що базовими моментами для блокування загроз компрометації чутливої інформації та маніпуляцій з даними поряд з забезпеченням високих криптографічних якостей алгоритму шифрування необхідними заходами є надійна ідентифікація пристроїв, безпечне управління ключами та скорочення кількості критичних криптографічних параметрів, що зберігаються на ПООР (безумовно краще повне виключення такої можливості).

До початку сеансу захищеного обміну, необхідно згенерувати на ПООР унікальні параметри для функціонування алгоритму: ключ K, синхромаркер S, та підстановку заміни X.

Звернемо увагу, вимога безпечного управління ключами шифрування корелюється з вимогою скорочення кількості критичних криптографічних параметрів, що зберігаються на ПООР. Дійсно, якщо для формування спільного сеансового ключу припустити використання симетричного алгоритму шифрування, то це потребує використання транспортного ключу для шифрування сеансових ключів під час передавання їх через незахищене середовище, а це породжує нову доволі складну проблему – надійний захист від несанкціонованого доступу до транспортного ключу на ПООР.

Застосування же асиметричних криптоалгоритмів хоча і призводить до певного навантаження на ПООР, але можливо відміти, наступне:

– формування сеансових ключів – це подія, яка відбувається відносно рідкісне, при цьому для відновлення сеансів зв’язку, що були розірвані внаслідок перешкод і збоїв обладнання припустимо не змінювати ключ, а генерувати новий синхромаркер;
– шифрування даних за допомогою асиметричних алгоритмів (так звана процедура – цифровий конверт) працює достатньо швидко, особливо в разі застосування перетворень, що базуються на еліптичних кривих [35, 100]. При цьому зворотна процедура – розшифрування ключу має відбуватись на більш потужному обчислювальному пристрої;
– сертифікати відкритих ключів у цьому випадку сприяють реалізації процедур ідентифікації.

З урахуванням зроблених зауважень захищений протокол взаємодії ПООР і хосту в мережі IoT має включати три фази:

– реалізація процедури ідентифікації і передача ПООР сертифікату відкритого ключу хоста;
– генерація сеансового ключу на ПООР, його шифрування за допомогою асиметричного алгоритму [107] та передачу його на хост;
– потокове шифрування даних на ПООР з використанням запропонованої модифікації алгоритму A5/1. За необхідності, має

відбуватись процедура відновлення зв'язку після короткочасних збоїв з використанням наявного сеансового ключу та нового випадкового синхромаркера.

Час існування сеансового ключу має бути обмеженим виходячи з умов експлуатації мережі.

Позначимо:

$Id1$ – ідентифікатор шлюзу

$Id2$ – ідентифікатор ПООР

S – синхромаркер

$C(X.509)$ – сертифікат відкритого ключа

K_e – відкритий ключ ДСТУ 9041

K_d – секретний ключ ДСТУ 9041

K_c – сеансовий ключ А5/1

$E(.,.)$ – алгоритм ДСТУ 9041

$M = (m_1, m_2, \dots, m_l)$ – послідовність відкритого повідомлення ПООР

$C = (c_1, c_2, \dots, c_l)$ – послідовність шифрованого тексту $C = A5\text{–}128(M, K_c)$

X – підстановка заміни – перший рядок латинського квадрата $L(X)$

$CRC32$ – контрольна сума

Загальний криптографічний протокол матиме вигляд (табл. 4.1).

Таблиця 4.1

Криптографічний протокол із застосуванням алгоритму А5-128

Крок	ПООР	Крок	Шлюз
1	Отримує $C(X.509)$ та перевіряє $Id1$	1	Надсилає $C(X.509)$
2	Генерує K_c, X, S $W = K_c \,\|\|X\|\|Id1\|\|Id2\|\|CRC32$	2	Отримує V та розшифровує секретним ключем $W = E^{-1}(V, K_d)$
3	Надсилає $V = E(W, K_e)$	3	Перевіряє ідентифікатори $Id1$ та $Id2$ та контрольну суму $CRC32$

4	Шифрує повідомлення $C = A5\text{–}128(M, K_c, S)$ та надсилає $C' = C\|\|S$	4	Отримує C', ініціалізовує синхромаркер та розшифровує повідомлення.

Звернемо увагу, у випадку надсилання великої кількості коротких повідомлень, передавати кожного разу новий синхромаркер може бути недоцільно з міркувань швидкодії. Тоді зміни протоколу будуть в наступному (рис. 4.1):

Крок 4. ПООР передає синхромаркер S з вказівкою шлюзу зберігати поточний стан регістрів, після чого шифрує повідомлення та передає C.

Крок 4. Шлюз отримує синхромаркер та вказівку і відслідковує стан регістрів та розшифровує кожне наступне C.

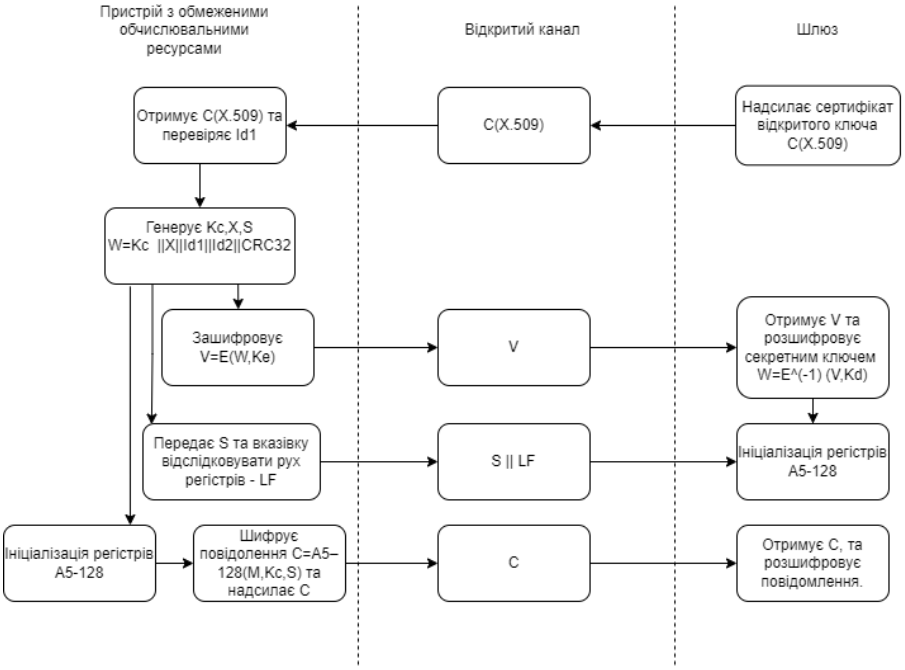

Рис. 4.1. Криптографічний протокол A5-128

Для впровадження алгоритму A5-128 було обрано один із протоколів безпроводового інформаційного обміну, що широко застосовується в IoT застосунках ESB, що також використовується сімейством чипів радіо зв’язку nRF24L01.

ESB представляє собою протокол передачі даних на основі пакетів. Він включає автоматичне формування та синхронізацію пакетів, автоматичне підтвердження та повторну передачу пакетів. ESB дозволяє створити реалізацію високоефективного зв'язку з високою продуктивністю за низьку ціну з використанням мікроконтролерів низької вартості. Як зазначено в [49], мережева топологія зв’язку ESB розроблена без будь-якого вбудованого шифрування чи процесу забезпечення безпеки за замовчуванням.

Пакет ESB включає в себе поле преамбули, поле адреси, поле керування пакетом, поле корисного навантаження та поле (циклічного надлишкового коду). На рис. 4.2 представлено формат пакета, де старший біт розташований ліворуч.

Преамбула 1 байт	Адреса 3-5 байт	Керування пакетом 9 біт	Корисне навантаження 0 - 32 байт	CRC 1-2 байти

Рис. 4.2 Вигляд пакету Enhanced ShockBurst за замовчуванням

Преамбула – це послідовність бітів, яка використовується для виявлення рівнів 0 та 1 на приймачі. Преамбула складається з одного байта і може бути або 01010101, або 10101010. Якщо перший біт у адресі дорівнює 1, преамбула автоматично встановлюється в 10101010, а якщо перший біт - 0, то преамбула автоматично встановлюється в 01010101. Це необхідно для забезпечення достатньої кількості переходів у преамбулі для стабілізації роботи приймача.

Адреса – це адреса для приймача. Адреса забезпечує виявлення пакетів приймачем. Поле адреси може бути налаштовано на довжину від 3 до 5 байт, адресу можна змінювати в процесі роботи. Зміна адреси за певним алгоритмом в процесі роботи, надає захист від DOS атак.

Поле керування пакетом містить поле де вказується довжина корисного навантаження за допомогою 6 біт, поле ідентифікації пакету з 2 біт і прапор NO_ACK з 1 біта, що вказує на автоматичне підтвердження.

Корисне навантаження – це визначений користувачем вміст пакету. Він може бути від 0 до 32 байтів і передається в прямому ефірі під час завантаження (без змін) на пристрій.

CRC – це механізм виявлення помилок у пакеті. Він може складати 1 або 2 байти та обчислюється за адресою, полем керування пакетом і корисним навантаженням.

У роботах [7, 74] пропонуються певні механізми забезпечення захисту інформації з використанням протоколу ESB. Недоліком цих робіт є використання вимогливих до обчислювальних ресурсів алгоритмів як AES та геш-функцій KMAC та SipHash. Як висновок [74] використання таких алгоритмів потребує дуже багато часу на шифрування даних.

Для забезпечення захисту інформації, що передається таким протоколом, збільшення швидкості шифрування та досягнення мети використання меншої кількості обчислювальних ресурсів для забезпечення достатнього рівня захисту вирішено впровадити модифікований алгоритм шифрування A5-128.

Полям пакету яким можна задати певне значення для забезпечення захисту є адреса та корисне навантаження. Оскільки інші поля використовуються як керуючі для роботи протоколу.

Основним методом забезпечення криптозахисту даного протоколу буде шифрування корисного навантаження. Звернемо увагу, розмір корисного навантаження одного пакету обмежений розміром 32 байти, а розмір V, що передається на етапі ініціалізації очевидно буде більшим, в такому випадку перед передачею V, ПООР повідомляє шлюз про розмір повідомлення V: N. Це стосується також всіх повідомлень розмір яких більше за 32 байти. Таким чином після застосування алгоритму A5-128 пакет матиме вигляд як на рис. 4.3.

Преамбула 1 байт	Адреса 3-5 байт	Керування пакетом 9 біт	Захищене корисне навантаження	CRC 1-2 байти

Рис. 4.3 Вигляд пакету Enhanced ShockBurst з шифруванням

Реалізація системи буде проводитись на основі методу криптографічного захисту інформації, що передається відкритими каналами зв'язку ПООР розробленому в 3 розділі. Загальна реалізація побудована на модифікованому алгоритмі, конкретні частини алгоритму розроблялись з урахуванням особливостей цільової платформи. Реалізація алгоритму була реалізована на мові С, в її варіанті представленому в середовищі розробки Arduino ide.

Для імітації корисних даних та оцінки швидкодії роботи алгоритму та споживання ресурсів було введено масив даних розміром 64 біти.

На стороні шлюзу також було реалізовано модифікований алгоритм, з механізмами збереження стану зсувних регістрів. На шлюзі використовується операційна система від Cisco, яка була обрана через те, що вона надає зручний доступ до програмування мікрокомп’ютера на мові Python, та є зручним інструментом для швидкого прототипування.

На стороні шлюзу повинні бути реалізовані загалом такі само компоненти методу як і на обмеженому пристрої, з урахуванням побудованого протоколу.

4.3. Оцінка ефективності алгоритму A5-128 на пристроях з обмеженими обчислювальними ресурсами

Для визначення ефективності модифікованого алгоритму, необхідно оцінити його за різними метриками на пристроях класу С0, використання яких обґрунтоване в розділі 2, якщо за різними параметрами алгоритм буде більш

ефективним від існуючих рішень, модель можна вважати ефективною. Для порівняння розробленої моделі були відібрані результати аналізу з розділу 2, відібрані алгоритми які показали себе найкраще за тими чи іншими показниками: HIGHT – як найшвидший алгоритм, XTEA – найзбалансованіший та Ascon – переможець конкурсу та запланований до стандартизації організацією NIST.

Загалом алгоритми будуть порівнюватись за тими ж само критеріями, що були визначені в розділі 2 та декілька додаткових (табл. 4.2).

Таблиця 4.2

Критерії оцінки алгоритму та інструменти вимірювання

Характеристика	Одиниці вимірювання	Інструмент вимірювання
Розмір ключа	Біти	Специфікація алгоритму
Розмір блоку тексту	Біти	Специфікація алгоритму
Використання ПЗП	Байти	Середовище програмування Arduino
Використання ОЗП	Байти	Середовище програмування Arduino
Швидкість шифрування	Байти/секунду	Програмування та обчислення
Затримка шифрування	Циклів/блок	Програмування та обчислення
Споживана потужність шифрування за секунду	Джоулі/секунду	Датчик струму Vernier та лабораторний комплекс LabQuest2
Споживана потужність за біт	Джоулі/біт	Датчик струму Vernier, лабораторний комплекс LabQuest2 та обчислення

Середовище розробки Arduino використовується для завантаження програмного коду на плату Arduino, по завершенню цього процесу відображаються такі параметри:

- використання ПЗП: Вимірюється кількість пам’яті, що необхідна для зберігання програми;

- використання ОЗП: Вимірюється скільки пам’яті займають локальні змінні, та вказується скільки залишається вільної пам’яті для динамічних структур.

Для декількох тестів буде проведено декілька вибірок, з яких буде взято середнє арифметичне. Це необхідно для отримання більш точного результату, оскільки при роботі контролеру можуть впливати різні фактори, такі як ступінь відхилення точності внутрішнього годинника від тактового генератора, переривання, перегрів, і т.д.

Такі кроки як генерування ключа, синхромаркера та обмін секретними параметрами, не будуть враховуватись в цій оцінці, через те, що це доволі рідкісна подія та її вплив на продуктивність є незначним. Окрім цього оцінка буде проводитись лише на ATmega328p, через те, що він був визначений раніше як пристрій класу C0, та експериментальним шляхом визначено, що існуючі рішення для нього є неефективними.

Розмір ключа та розмір блоку. Для порівняння цих двох характеристик були використані дані специфікації алгоритмів та характеристики розробленого алгоритму. Алгоритми, що застосовуються звичайних моделях, є блоковими вони мають вимоги до розміру ключа 128 біт, та у випадку з Ascon [24], ще 128 біт криптографічного нонс. Модифікований алгоритм також має розмір ключа 128 біт, та 128 біт синхромаркер який дозволяє відновити захищений зв’язок у випадку збоїв без необхідності генерування та надсилання нового ключа.

Кількість раундів. Як частина функціоналу алгоритму, сучасні шифри збільшують їхню безпеку (змішування та дифузія) за допомогою повторного виконання (n разів) функції раунду. У блокових шифрах вхід і вихід функції раунду рівні розміру блоку шифру в загальному випадку. Як стандартне правило, збільшення кількості раундів n підвищує рівень безпеки, тоді як зменшення кількості раундів відіграє значну роль у скороченні часу виконання шифрування і розшифрування, що є однією з сутностей КПЗВ NIST. Проте для модифікованого алгоритму, кількість раундів не відіграє ролі, оскільки

використовується потоковий шифр, тому порівнювати за цим критерієм є недоречно.

Використання ПЗП. Пам'ять тільки для читання або ПЗП – це енергонезалежна пам'ять. Дані, що зберігаються в ПЗП, зазвичай є скомпільованим кодом і таблицями, які не потребують змін. Такий розмір менше за розмір коду. Для оцінки мінімального розміру ПЗП з реалізації було видалено частини з передачею або виводом даних, оскільки це суттєво збільшує розмір за рахунок використання відповідних програмних бібліотек. Таким чином модифікований алгоритм, використовує значно менше ПЗП пам’яті від алгоритму Ascon, та використовує приблизно стільки ж пам’яті як решта алгоритмів (рис. 4.4).

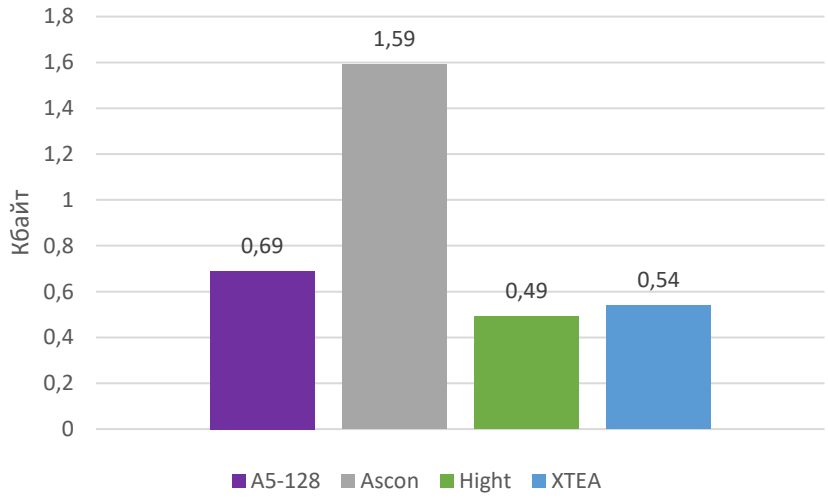

Рис. 4.4. Використання постійної пам’яті

Використання ОЗП. ОЗП – це короткочасна пам'ять, у якій дані зберігаються для обробки процесором. Дані зберігаються в ОЗП у формі кучі та стеку. Розмір використовуваної ОЗП на мікроконтролері становить 2 кБайти. Для визначення розміру використовувались вбудовані функції середовища розробки, а також оцінка вільної пам’яті програмними засобами в процесі виконання алгоритму. Оцінка проводилась тільки, щодо пам’яті, що використовується алгоритмом, решта функцій були заміряні окремо та

виокремлені від результату. За результатами аналізу визначено, що модифікований алгоритм використовує оперативну пам’ять на рівні з іншими алгоритмами, при цьому потребує менше оперативної пам’яті від найшвидшого серед протестований алгоритму HIGHT на 31.25% (рис. 4.5).

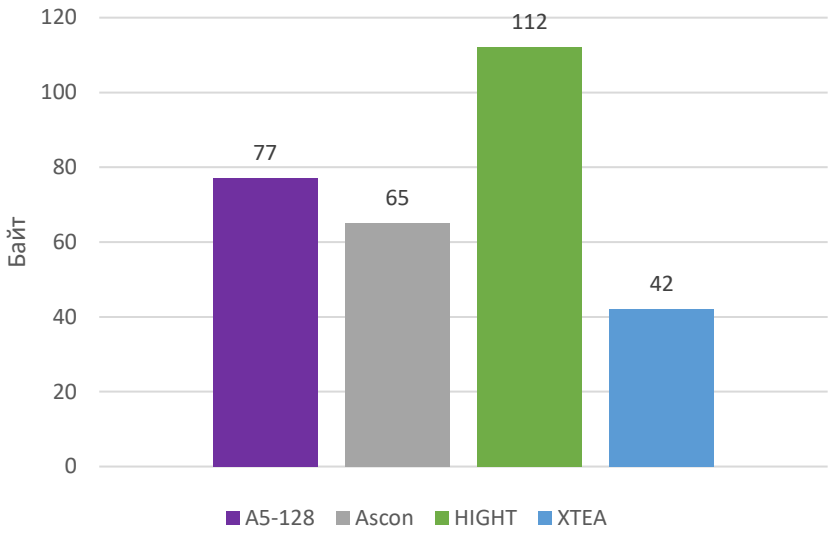

Рис. 4.5. Використання оперативної пам’яті

Швидкість шифрування або розшифрування. Швидкість виконання операції шифрування та розшифрування була визначена використанням відповідних функцій мови С для фіксування часу та тактів процесору, після чого була використана формула наведена раніше. Така швидкість модифікованого алгоритму була досягнута завдяки тому, що алгоритм використовує ефективні операції за модулем 256, що дозволяє виконувати їх ефективніше та реалізації зсуву без копіювання даних. Таким чином пропонована має на 30.70% вищу швидкість шифрування в порівнянні з найшвидшим алгоритмом виявленим в другому розділі HIGHT, та значно вищу швидкість ніж алгоритм Ascon, та XTEA на 8-бітному пристрої (рис. 4.6).

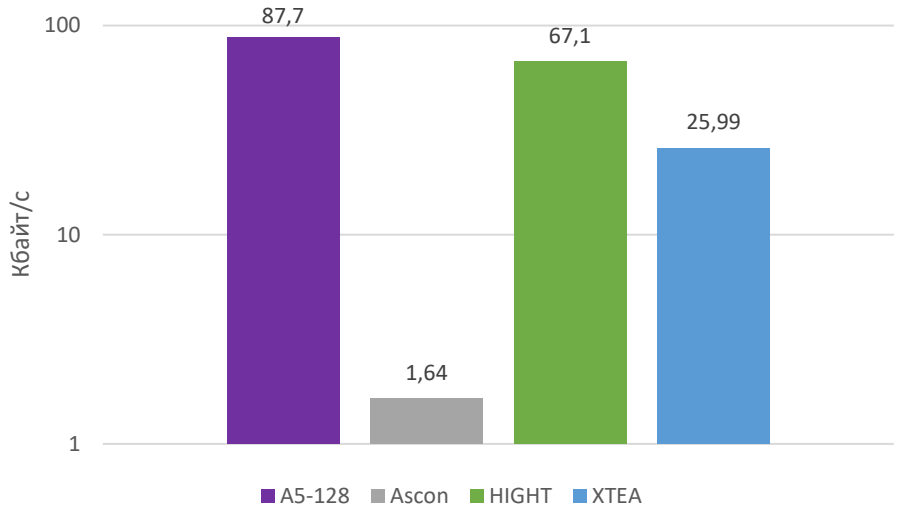

Рис. 4.6. Швидкість шифрування та розшифрування

Ґрунтуючись на цьому можна знайти *затримку шифрування або розшифрування*. Частота процесору 16 МГц, використовуючи отримані раніше дані, було визначено затримку для шифрування блоку тексту. Оскільки затримка прямо залежить від швидкості виконання шифрування/розшифрування, то за результатами аналізу було отримано «перевернутий» графік в порівнянні з попереднім, через те, що Ascon вимагає найбільше часу на шифрування відповідно він має і найбільшу затримку. Модифікований алгоритм криптографічного захисту, має найменшу затримку (рис. 4.7).

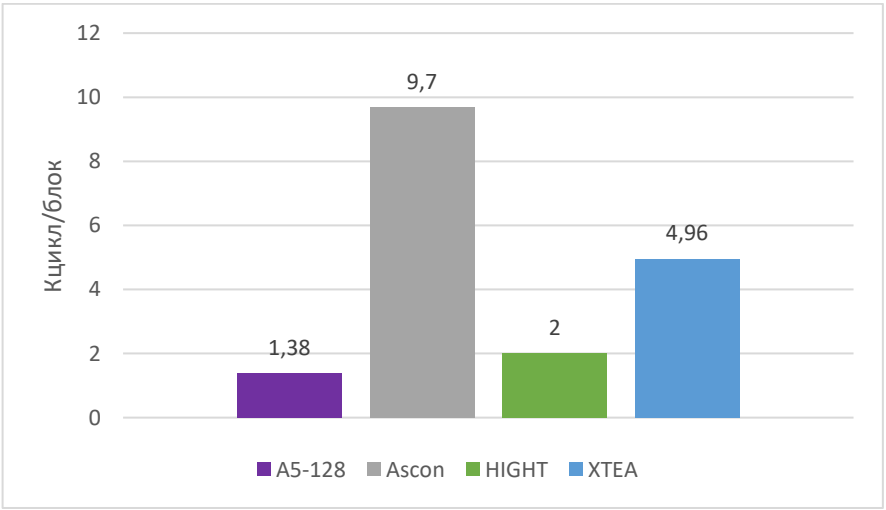

Рис. 4.7. Програмна затримка шифрування

Споживання енергії для шифрування за секунду. Витрата енергії є мірою використання електричного струму під час виконання операції алгоритму, вся спожита енергія при цьому є показником потужності. Для вимірювання енергії, що споживається був використаний лабораторний блок живлення, та датчик струму виробництва Vernier з лабораторним комплексом LabQuest2. Отримані результати демонструють низьке споживання енергії пристроєм при використанні модифікованого алгоритму на рівні найкращого результату виявленого в другому розділі та значно менше в порівнянні з рештою алгоритмів (рис. 4.8).

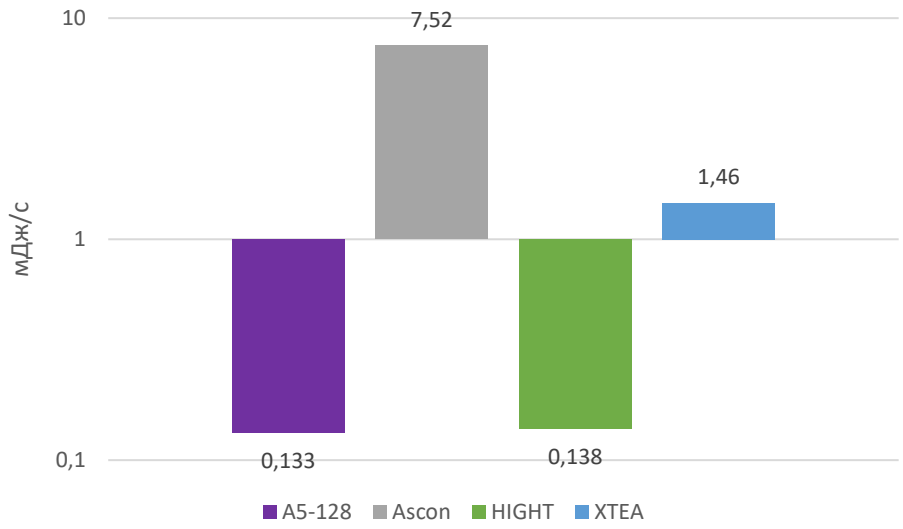

Рис. 4.8. Споживання енергії на шифрування за секунду

Споживання енергії на шифрування одного біта. Якщо аналізувати з точки зору спожитої енергії на біт (рис. 4.9) результат приблизно такий самий. Алгоритм Ascon демонструє надто велике споживання енергії як для швидкості шифрування яку він демонструє. При цьому A5-128, що реалізовано у розробленому методі, демонструє кращий результат через високу швидкість шифрування.

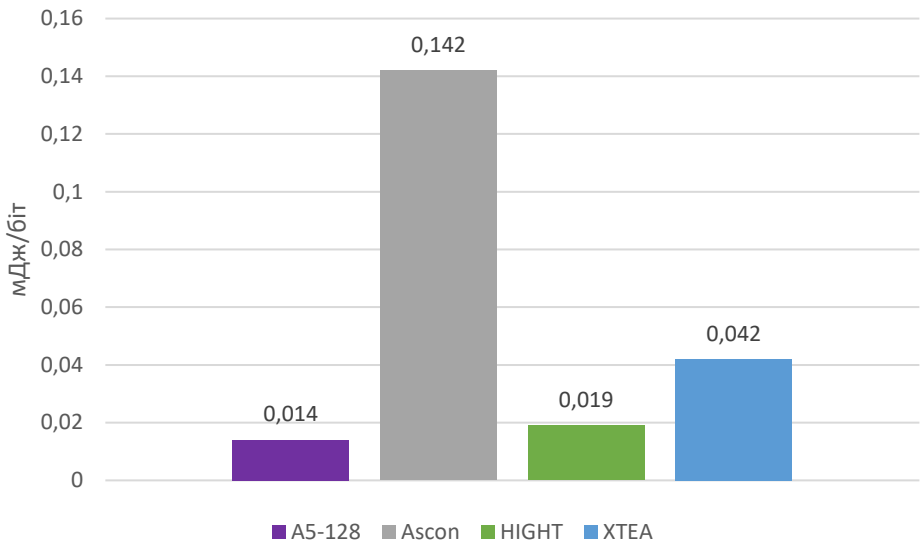

Рис. 4.9. Споживання енергії на шифрування одного біта

Окрім характеристик продуктивності, модифікований алгоритм забезпечує підвищену криптостійкість та імітостійкість шифрування завдяки застосуванню байтової обробки інформації та застосування вузла накладання шифру багатоалфавітної заміни.

Таким чином, проведений аналіз доводить ефективність методу криптографічного захисту на основі модифікованого алгоритму A5-128 на пристроях класу C0 в мережах IoT. Проте, варто зауважити, що алгоритм Ascon може бути ефективним на більш потужних пристроях 64-бітної архітектури для яких він був спроєктований.

Отож, модифікований криптографічний алгоритм, захисту інформації, що передається відкритими каналами зв'язку ПООР в мережах IoT дозволяє збільшити швидкість шифрування при використанні мінімуму обчислювальних ресурсів, при цьому забезпечує підвищений рівень криптостійкості та імітостійкості. Модифікований алгоритм підвищує захищеність ПООР та мереж де вони використовуються.

Загальні рекомендації щодо впровадження методу криптографічного захисту інформації, що передається відкритими каналами зв’язку в мережах IoT:

1. Дотримуватись криптографічного протоколу, в частині передачі ключа шифрування.

2. Враховувати особливості пристроїв на яких він розгортається, зокрема для забезпечення високої швидкодії.

3. За можливості передбачити додаткові джерела для формування ключа та синхромаркера для усунення наслідків фізичного впливу на АЦП.

4. Використання унікальних ідентифікаторів для надійної ідентифікації пристроїв в мережі.

5. Передбачити перевірку ключа та синхромаркера на предмет відхилення від рівномірного розподілення для виявлення аномальної поведінки при генерації критичних параметрів.

6. Проводити аналіз ефективності та результативності алгоритму для вдосконалення та оптимізації його використання.

Окрім цього, перед впровадженням методу необхідно також врахувати наступні рекомендації:

- забезпечити фізичний контур безпеки;
- у разі встановлення обладнання в умовах агресивного зовнішнього середовища надати перевагу використанню спеціалізованих корпусів для захисту пристроїв від природніх впливів;
- передбачити використання резервних джерел живлення;
- необхідно забезпечити регулярне оновлення параметрів системи;
- мати достатній рівень технічної підтримки для реалізації методу криптографічного захисту інформації.

Висновки до четвертого розділу

1. В результаті аналізу платформ для побудови прототипу рішення було використано мікроконтролер ATmega328, в якості обмеженого пристрою

класу C0, для якого пропоновані шифри є неефективними через 8-бітний процесор та малу кількість пам’яті. Raspberry Pi 3 Model B+ було використано в ролі шлюзу через достатні обчислювальні характеристики, наявність сучасних інтерфейсів для підключення до існуючих класичних мереж: Gigabit Ethernet, IEEE 802.11 b/g/n/ac (2.4GHz / 5GHz), та невелику собівартість для реалізації рішення.

2. В ході реалізації модифікованого алгоритму A5-128 вдосконалено стандартний протокол Shockburst безпроводового інформаційного обміну в мережі IoT. Криптографічний протокол враховує надійну ідентифікацію пристроїв, безпечне управління сеансовими ключами та скорочує кількість критичних криптографічних параметрів, що зберігаються на ПООР. Удосконалений протокол забезпечує криптографічно захищену передачу даних з високою імітостійкістю та механізмами відновлення сеансу зв’язку від ПООР до шлюза.

3. У результаті дослідження та проведення практичного експерименту щодо функціонування модифікованого алгоритму на пристроях класу C0 було виявлено, що алгоритм A5-128, використовує ОЗП та ПЗП пам’яті на рівні з іншими алгоритмами, при цьому має на 30,70% більшу швидкість шифрування ніж найшвидший з досліджених алгоритм Hight. Споживає на 3,62% менше енергії, та забезпечує підвищену криптостійкість та імітостійкість шифрування.

ВИСНОВКИ

В результаті дисертаційних досліджень, виконаних автором, вирішено актуальне наукове завдання, що полягає в розробці моделей та методів захисту інформації, що передається відкритими каналами зв'язку пристроями з обмеженими обчислювальними ресурсами в мережах інтернету речей за допомогою криптографічних перетворень, що забезпечують підвищений рівень конфіденційності, криптостійкості, імітостійкості та високу швидкість шифрування на основі модифікації стандартного криптографічного алгоритму A5/1. Зазначене наукове завдання має суттєве значення для забезпечення безпеки мереж IoT, а також підвищує ефективність шифрування даних на пристроях класу C0 з 8-бітними процесорами. Отримані показники ефективності роблять результати дослідження актуальними та пріоритетними.

В дисертаційному дослідженні отримані такі основні результати:

1. На підставі проведеного аналізу виявлено, що системи типу M2M (інтернету речей) становитимуть половину всіх глобальних підключень пристроїв та, що складає близько 14,7 млрд пристроїв. Значна частина таких пристроїв мають обмежену кількість обчислювальних ресурсів таких як частота процесору, об’єм оперативної та постійної пам’яті, живляться від акумуляторних джерел відповідно мають обмеження в споживанні енергії. Зважаючи на такі характеристики вони часто не можуть використовувати стандартні алгоритми криптографічного захисту інформації, проте вони повинні бути захищені. Як наслідок, необхідність в розробці алгоритмів захисту інформації для використання пристроями інтернету речей. При аналізі поточного стану досліджень в цій області було виявлено, що існують спеціально розроблені для таких систем – криптографічні алгоритми. Однак досліджені алгоритми мають певні недоліки: більшість алгоритмів розроблена для використання на пристроях класу C1 та C2 які мають 64-бітні процесори, що робить їх неефективними з точки зору швидкодії та споживання ресурсів

на пристроях класу С0, з 8- та 16-бітними процесорами. Окрім цього існує зауваження щодо прозорості їхнього проєктування, зокрема недостатність інформації щодо умов їх безпечного застосування та управління криптографічними ключами (окрім національних). Основне протиріччя, яке лежить в основі наукового дослідження полягає в тому, що існує необхідність захисту даних на ПООР класу С0, використовуючи мінімум обчислювальних ресурсів та енергії. Отже, актуальним є наукове завдання, щодо розробки моделей та методів захисту інформації, що передається відкритими каналами зв'язку ПООР в мережах IoT за допомогою криптографічних перетворень, що забезпечують підвищений рівень конфіденційності, криптостійкості, імітостійкості та високу швидкість шифрування на основі модифікації стандартного криптографічного алгоритму А5/1.

2. Вперше запропоновано метод криптографічного захисту інформації в мережі IoT на основі модифікованого алгоритму А5/1, що забезпечує підвищену стійкість шифрування та імітостійкість на основі застосування байтової обробки інформації та застосування вузла накладання шифру на основі змінного латинського квадрату. Алгоритм має високу швидкодію, яка на 30.70% більше ніж у існуючих алгоритмів для IoT.

3. Вдосконалено стандартний протокол Shockburst безпроводового інформаційного обміну в мережі з метою безпечного формування сеансових ключів та забезпечення криптографічно захищеної передачі даних від пристроїв з обмеженими обчислювальними ресурсами до шлюзу.

4. Подальшого розвитку набула модель загроз для побудови системи захисту інформації, що обробляється пристроями з обмеженими обчислювальними ресурсами в мережі інтернету речей.

5. Проведено оцінку функціонування модифікованого алгоритму, а саме шифруючої послідовності шляхом використання набору статистичних тестів. У всіх тестах алгоритм формування шифруючої послідовності показав результат згідно якого послідовність може вважатись випадковою та рівномірно розподіленою.

6. У результаті дослідження та проведення практичного експерименту щодо функціонування модифікованого алгоритму на пристроях класу С0 було виявлено, що алгоритм А5-128, використовує ОЗП та ПЗП пам’яті на рівні з іншими алгоритмами, при цьому має на 30.70% більшу швидкість шифрування ніж найшвидший з досліджених алгоритм Hight. Споживає на 3.62% менше енергії, та забезпечує підвищену криптостійкість та імітостійкість шифрування.

7. Розроблено рекомендації, щодо впровадження методу криптографічного захисту інформації, що передається відкритими каналами зв'язку пристроями з обмеженими обчислювальними ресурсами в мережах IoT та описаний алгоритм роботи розробленої моделі.

8. Таким чином, мета дослідження яка полягає у забезпеченні безпеки інформаційних ресурсів в мережах інтернету речей, включаючи їх конфіденційність і цілісність, за рахунок розробки молей і методів криптографічного захисту інформації, що передається пристроями з обмеженими обчислювальними ресурсами досягнута та всі часткові завдання вирішено повністю.

9. Основні наукові результати дослідження реалізовані у практичній діяльності в ТОВ «2Д3Д», та в ТОВ «Технологічні ІТ рішення», які використані для удосконалення функціонування мікроконтролерів та підвищенню рівня захисту інформації та в освітній процес Київського столичного університету імені Бориса Грінченка.

10. Напрями подальших досліджень в зазначеній галузі можуть ґрунтуватись на вдосконаленні процедур генерації випадкових параметрів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Abomhara, M., & Koien, G. M. (2014). Security and privacy in the internet of things: Current status and open issues. 2014 International Conference on Privacy and Security in Mobile Systems (PRISMS), pp. 1-8. http://dx.doi.org/10.1109/PRISMS.2014.6970594.

2. Al-Shargabi, B., & Dar Assi, A. (2023). A modified lightweight DNA-based cryptography method for Internet of Things devices. Expert Systems, 40(6), e13270. https://doi.org/10.1111/exsy.13270

3. Arduino. Retrieved 2020, from https://www.arduino.cc/

4. Asplund, M., & Nadjm-Tehrani, S. (2016). Attitudes and Perceptions of IoT Security in Critical Societal Services. IEEE Access, 4, 1-1. https://doi.org/10.1109/ACCESS.2016.2560919.

5. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A Survey. Computer Networks, 54(15), 2787-2805. https://doi.org/10.1016/j.comnet.2010.05.010.

6. Avanzi, R. (2017). The QARMA Block Cipher Family. Almost MDS Matrices Over Rings With Zero Divisors, Nearly Symmetric Even-Mansour Constructions With Non-Involutory Central Rounds, and Search Heuristics for Low-Latency S-Boxes. IACR Transactions on Symmetric Cryptology, 2017(1), 4–44. https://doi.org/10.13154/tosc.v2017.i1.4-44

7. Ayati, A., & Naji, H. (2023). A security mechanism for Enhanced ShockBurst wireless communication protocol using nRF24L01. https://doi.org/10.21203/rs.3.rs-3777984/v1

8. Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., & Regazzoni, F. (2015). Midori: A Block Cipher for Low Energy (Extended Version). Cryptology ePrint Archive, Paper 2015/1142. Retrieved from https://eprint.iacr.org/2015/1142

9. Bassham, L. E. III, Çalık, Ç., McKay, K. A., Mouha, N., & Sönmez Turan, M. (April 26, 2017). (Draft) Profiles for the Lightweight Cryptography

Standardization Process. National Institute of Standards and Technology Draft Whitepaper. Retrieved from https://csrc.nist.gov/publications/detail/white-paper/2017/04/26/profiles-for-lightweightcryptography-standardization-process/archive.

10. Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2012). Permutation-based encryption, authentication, and authenticated encryption. Retrieved from https://api.semanticscholar.org/CorpusID:192955

11. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G. (2011). Keccak Specifications. Submission to NIST (Round 3). Retrieved from http://keccak.noekeon.org

12. Biryukov, A. (2011). DES-X (or DESX). In: van Tilborg, H.C.A., Jajodia, S. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA. https://doi.org/10.1007/978-1-4419-5906-5_570

13. Borghof, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., ... Yalçın, T. (2012). PRINCE—A Low-Latency Block Cipher for Pervasive Computing Applications. International Conference on the Theory and Application of Cryptology and Information Security, Beijing, 208-225. https://doi.org/10.1007/978-3-642-34961-4_14.

14. Bormann, C., Ersue, M., & Keranen, A. (2014). Terminology for Constrained-Node Networks. Internet Engineering Task Force (IETF). ISSN: 2070-1721. from https://tools.ietf.org/html/rfc7228

15. Brown, R. G. (2004). Dieharder: A Random Number Test Suite, Version 3.31.1. Retrieved from http://rgbrown.org/General/general.php

16. Buchanan, W. J., Li, S., & Asif, R. (2017). Lightweight cryptography methods. Journal of Cyber Security Technology, 1(3-4), 187-201. https://doi.org/10.1080/23742917.2017.1384917.

17. Buhantsov, A. D., Sadjiid, A. Yu., Ustinov, A. N., & Rodionov, C. V. (2021). Research of speech encryption reliability in GSM mobile communication technology. Research result. Information technologies, 6(2), 9-17. DOI: 10.18413/2518-1092-2021-6-2-0-2

18. Chernenko, R., Anosov, A., Kyrychok, R., Brzhevska, Z., & Spasiteleva, S. (2022). Encryption Method for Systems with Limited Computing Resources. CEUR Workshop Proceedings, 3288, pp. 142-148. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85143827975&partnerID=40&md5=fc5b960373acefb92e4755f0a571afb2

19. CISCO. (2014). The Internet of Things reference model. Retrieved from https://dl.icdst.org/pdfs/files4/0f1d1327c5195d1922175dd77878b9fb.pdf

20. Daemen, J., & Rijmen, V. (1998). AES Proposal: Rijndael. AES Round 1 Technical Evaluation CD1: Documentation. National Institute of Standards and Technology. Retrieved from http://www.nist.gov/aes.

21. Dobraunig, C., Eichlseder, M., Mangard, S., Mendel, F., Mennink, B., & Primas, R. (2021). ISAP v2.0. Submission to NIST LWC Project. Retrieved from https://isap.iaik.tugraz.at

22. Dobraunig, C., Eichlseder, M., Mangard, S., Mendel, F., Mennink, B., Primas, R., & Unterluggauer, T. (May 17, 2021). ISAP v2.0 Submission to NIST. Retrieved 2023 from https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/isap-spec-final.pdf

23. Dobraunig, C., Eichlseder, M., Mendel, F., & Schläffer, M. (2021). ASCON v1.2: Lightweight Authenticated Encryption and Hashing. Journal of Cryptology, 34(1), 33. https://doi.org/10.1007/s00145-021-09398-9

24. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M. (2021). Ascon v1.2. Submission to NIST. Retrieved 2023 from https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/ascon-spec-final.pdf

25. eBACS: ECRYPT Benchmarking of Cryptographic Systems. Bench. (2019). Retrieved 2023, from https://bench.cr.yp.to

26. El-hajj, M., Mousawi, H., & Fadlallah, A. (2023). Analysis of Lightweight Cryptographic Algorithms on IoT Hardware Platform. Future Internet, 15, 54. https://doi.org/10.3390/fi15020054

27. European Commission. (2004). Final report of European project number IST-1999-12324, named New European Schemes for Signatures, Integrity, and Encryption. NESSIE public report D20. NESSIE Security Report. Springer-Verlag. Retrieved from http://cryptonessie.org.

28. Glukhov, M., Elizarov, V., & Nechaev, A. (2003). Algebra, vol. 2. Gelios APB. ISBN 8-85338-072-2.

29. Granjal, J., Monteiro, E., & Sa Silva, J. (2015). Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. IEEE Communications Surveys & Tutorials, 17, 1294-1312. https://doi.org/10.1109/COMST.2015.2388550.

30. Grechaninov, V., Hulak, H., Hulak, E., Skladannyi, P., & Sokolov, V. (2021). Decentralized Access Demarcation System Construction in Situational Center Network Cybersecurity Providing in Information and Telecommunication Systems II. Cybersecurity Providing in Information and Telecommunication Systems II, 3188(2), 197-206. ISSN 1613-0073. Retrieved from https://ceur-ws.org/Vol-3188/paper18.pdf

31. Grechaninov, V., Hulak, H., Sokolov, V., Skladannyi, P., & Korshun, N. (2022). Formation of Dependability and Cyber Protection Model in Information Systems of Situational Center. Retrieved from https://ceur-ws.org/Vol-3149/paper11.pdf

32. Greenberger, M. (1965). Method in Randomness. Communications of the ACM, 8(3), 177–179. https://doi.org/10.1145/363791.363827

33. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. Future Generation Computer Systems, 29, 1645-1660. https://doi.org/10.1016/j.future.2013.01.010.

34. Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I., & Manifavas, H. (2018). A review of lightweight block ciphers. Journal of Cryptographic Engineering, 8, 1-44. https://doi.org/10.1007/s13389-017-0160-y

35. Hoffstein, J., Pipher, J., & Silverman, J. H. (2014). An Introduction to Mathematical Cryptography. Springer. ISBN 978-1-4939-1711-2. https://doi.org/10.1007/978-1-4939-1711-2

36. Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B.-S., ... Lee, J. (2006). HIGHT: A New Block Cipher Suitable for Low-Resource Device. У L. Goubin & M. Matsui (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2006 (Lecture Notes in Computer Science, Vol. 4249, pp. 1-14). Springer. https://doi.org/10.1007/11894063_4

37. Hong, S., Hong, D., Ko, Y., Chang, D., Lee, W., & Lee, S. (2004). Differential Cryptanalysis of TEA and XTEA. In JI. Lim & DH. Lee (Eds.), Information Security and Cryptology - ICISC 2003 (Lecture Notes in Computer Science, Vol. 2971). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-24691-6_30

38. Hosseinzadeh, J., & Ghaemi Bafghi, A. (2017). Software Implementation And Evaluation Of Lightweight Symmetric Block Ciphers Of The Energy Perspectives And Memory. Retrieved from http://arxiv.org/abs/1706.03909

39. Hulak, H., Skladannyi, P., Sokolov, V., Hulak, E., & Korniiets, V. (2022). Dynamic model of guarantee capacity and cyber security management in the critical automated systems. In 2nd International Conference on Conflict Management in Global Information Networks (CMiGiN 2022), 3530 (pp. 102-111). Retrieved from https://ceur-ws.org/Vol-3530/paper11.pdf

40. Ibrahim, N., & Agbinya, J. (2022). A Review of Lightweight Cryptographic Schemes and Fundamental Cryptographic Characteristics of Boolean Functions. Advances in Internet of Things, 12, 9-17. https://doi.org/10.4236/ait.2022.121002.

41. ISO/IEC. (2012). Information technology — Security techniques — Lightweight cryptography — Part 2: Block ciphers. Retrieved from https://www.iso.org/standard/56552.html

42. Jean, J., Nikolić, I., Peyrin, T., Wang, L., & Wu, S. (2014). Security Analysis of PRINCE. In: Moriai, S. (eds) Fast Software Encryption. FSE 2013.

Lecture Notes in Computer Science(), vol 8424. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-43933-3_6

43. Joint Task Force Transformation Initiative. (2012). NIST Special Publication 800-30 Revision 1: Guide for Conducting Risk Assessments. https://doi.org/10.6028/NIST.SP.800-30r1

44. Juels, A., & Weis, S. A. (2005). Authenticating pervasive devices with human protocols. In Proc. 25th Annu. Int. Cryptol. Conf. (pp. 293–308). Berlin, Germany: Springer. https://link.springer.com/chapter/10.1007/11535218_18

45. Katagi, M., & Moriai, S. (2012). Lightweight Cryptography for the Internet of Things. Sony Corporation. from https://www.iab.org/wp-content/IAB-uploads/2011/03/Kaftan.pdf.

46. Kolisnyk, M. (2021). Vulnerability analysis and method of selection of communication protocols for information transfer in Internet of Things systems. Radioelectronic and Computer Systems. https://doi.org/10.32620/reks.2021.1.12

47. Koo, B., Hong, D., & Kwon, D. (2011). Related-Key Attack on the Full HIGHT. In K. H. Rhee & D. Nyang (Eds.), Information Security and Cryptology - ICISC 2010 (Lecture Notes in Computer Science, Vol. 6829, pp. 4). Springer. https://doi.org/10.1007/978-3-642-24209-0_4

48. Kristinsson, B. (2011). Ardrand: The Arduino as a Hardware Random-Number Generator. Retrieved from https://api.semanticscholar.org/CorpusID:195592641

49. Kulasekara, V., Balasooriya, S., Chandran, J., & Kavalchuk, I. (2019). Novel Low-Power NRF24L01 Based Wireless Network Design for Autonomous Robots. In 2019 25th Asia-Pacific Conference on Communications (APCC) (pp. 342-346). Ho Chi Minh City, Vietnam. https://doi.org/10.1109/APCC47188.2019.9026452.

50. Kuzminykh, I., Carlsson, A., Yevdokymenko, M., & Sokolov, V. (2019). Investigation of the IoT Device Lifetime with Secure Data Transmission. https://doi.org/10.1007/978-3-030-30859-9_2

51. Leander, G., Paar, C., Poschmann, A., & Schramm, K. (2007). New Lightweight DES Variants. In 14th Annual Fast Software Encryption Workshop (FSE 2007), Luxembourg, March 26-28, 196-210. https://doi.org/10.1007/978-3-540-74619-5_13.

52. L'Ecuyer, P., & Simard, R. (2007). TestU01: A C library for empirical testing of random number generators. ACM Transactions on Mathematical Software, 33(4), Article 22. https://doi.org/10.1145/1268776.1268777

53. Liebl, S. (2023). Threat Modelling for Internet of Things Devices. Retrieved from https://www.researchgate.net/publication/369488078_Threat_Modelling_for_Internet_of_Things_Devices

54. Lightweight Cryptography. National Institute of Standards and Technology. (2017). Retrieved 2022, from https://csrc.nist.gov/Projects/Lightweight-Cryptography

55. Lim, Y. I., Lee, J. H., You, Y., & Cho, K. R. (2009). Implementation of HIGHT cryptic circuit for RFID tag. IEICE Electronics Express, 6(4), 180-186. https://www.jstage.jst.go.jp/article/elex/6/4/6_4_180/_pdf

56. Lo, O., Buchanan, W., & Carson, D. (2018). Correlation Power Analysis on the PRESENT Block Cipher on an Embedded Device. 1-6. https://doi.org/10.1145/3230833.3232801

57. Massey, J.L., Maurer, U., Wang, M. (1988). Non-Expanding, Key-Minimal, Robustly-Perfect, Linear and Bilinear Ciphers. In: Chaum, D., Price, W.L. (eds) Advances in Cryptology — EUROCRYPT' 87. EUROCRYPT 1987. Lecture Notes in Computer Science, vol 304. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-39118-5_22

58. McKay, K. A., Bassham, L. E. III, Sönmez Turan, M., & Mouha, N. (2017). Report on Lightweight Cryptography. National Institute of Standards and Technology, Gaithersburg, MD. NIST Internal Report (IR) 8114. https://doi.org/10.6028/NIST.IR.8114.

59. Mhaibes, H. I., Abood, M. H., & Farhan, A. (2022). Simple Lightweight Cryptographic Algorithm to Secure Embedded IoT Devices. International Journal of Interactive Mobile Technologies (iJIM), 16(20), 98–113. https://doi.org/10.3991/ijim.v16i20.34505

60. Moshenchenko, M., Zhurakovskyi, B., Poltorak, V., Bondarchuk, A., & Korshun, N. (2021). Optimization Algorithms of Smart City Wireless Sensor Network Control. CEUR Workshop Proceedings, 3188, 32-42. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-85137148248&partnerID=40&md5=afdbc30eb5b9af653151a83cb7c9913d

61. Mouha, N. (2015). The Design Space of Lightweight Cryptography. Cryptology ePrint Archive, Paper 2015/303. Retrieved from https://eprint.iacr.org/2015/303

62. National Institute of Standards and Technology. (2012). FIPS PUB 180-4: Secure Hash Standard. Federal Information Processing Standards Publication 180-4. U.S. Department of Commerce. Retrieved from http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

63. National Institute of Standards and Technology. (2023). NIST Selects 'Lightweight Cryptography' Algorithms to Protect Small Devices. Retrieved 2024, from https://www.nist.gov/news-events/news/2023/02/nist-selects-lightweight-cryptography-algorithms-protect-small-devices

64. NIST. (2018). Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process. from https://csrc.nist.gov/csrc/media/Projects/lightweight-cryptography/documents/final-lwc-submission-requirements-august2018.pdf

65. Nordic Semiconductor. Enhanced ShockBurst Documentation. Retrieved from https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/esb/index.html

66. Omrani, T., Rhouma, R., & Sliman, L. (2018). Lightweight Cryptography for Resource-Constrained Devices: A Comparative Study and

Rectangle Cryptanalysis. Third International Conference, ICDEc 2018, Brest, France, May 3-5, 2018, Proceedings. https://doi.org/10.1007/978-3-319-97749-2_8.

67. Onyshchenko, V., Negodenko, O., & Shevchenko, S. (2019). Models of Information Processing in IoT Networks on the Basis of Fundamental Trigonometric Splines. In 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine (pp. 613-616). doi: 10.1109/PICST47496.2019.9061424.

68. OWASP. "OWASP Top 10 2018: Internet of Things". Retrieved from https://owasp.org/www-pdf-archive/OWASP-IoT-Top-10-2018-final.pdf [accessed: 2022-08-26]

69. Özen, O., Varıcı, K., Tezcan, C., & Kocair, Ç. (2009). Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced Round PRESENT and HIGHT. In C. Boyd & J. González Nieto (Eds.), Information Security and Privacy. ACISP 2009 (Lecture Notes in Computer Science, Vol. 5594, pp. 7). Springer. https://doi.org/10.1007/978-3-642-02620-1_7

70. Pradhan, D., & Tun, H. (2022). Security Challenges: M2M Communication in IoT. Journal of Electrical Engineering and Automation, 4, 187-199. https://doi.org/10.36548/jeea.2022.3.006

71. Precedence Research. (2023). Microcontroller (MCU) Market Regional Outlook 2023 - 2032. Precedence Research. Retrieved from https://www.precedenceresearch.com/microcontroller-mcu-market

72. Ragab, A. A. M., Madani, A., Wahdan, A. M., & Selim, G. M. I. (2021). Design, Analysis, and Implementation of a New Lightweight Block Cipher for Protecting IoT Smart Devices. Journal of Ambient Intelligence and Humanized Computing. https://doi.org/10.1007/s12652-020-02782-6.

73. Rahman, M. S., Karnik, S., & Sarangerel, S. (2022). Lightweight Cryptography. MIT Course Project. Retrieved from https://courses.csail.mit.edu/6.857/2022/projects/Shahir-Rahman-Karnik-Sarangerel.pdf.

74. Rivera, D., Garcia, A., Martín-Ruiz, M., Alarcos, B., Velasco, J., & Oliva, A. (2019). Secure Communications and Protected Data for an Internet of Things Smart Toy Platform. IEEE Internet of Things Journal, 1-1. https://doi.org/10.1109/JIOT.2019.2891103.

75. Rizvi, S., Kurtz, A., Pfeffer, J., & Rizvi, M. (2018). Securing the Internet of Things (IoT): A Security Taxonomy for IoT. pp. 163-168. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00034.

76. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., ... Bassham, L. (2010). NIST SP 800-22 Rev. 1: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. National Institute of Standards and Technology Special Publication 800-22 Rev. 1. https://doi.org/10.6028/NIST.SP.800-22r1a

77. Sethi, P., & Sarangi, S. R. (2017). Internet of Things: Architectures, Protocols, and Applications. Journal of Electrical and Computer Engineering, 2017. https://doi.org/10.1155/2017/9324035.

78. Shannon, C. E. (1949). Communication theory of secrecy systems. The Bell System Technical Journal, 28(4), 656-715. https://doi.org/10.1002/j.1538-7305.1949.tb00928.x

79. Shelby, Z., & Bormann, C. (2009). 6LoWPAN: The Wireless Embedded Internet. November 2009, 244 pages. ISBN: 978-0-470-74799-5.

80. Shevchenko, O., Bondarchuk, A., Polonevych, O., Zhurakovskyi, B., & Korshun, N. (2022). Methods of the Objects Identification and Recognition Research in the Networks with the IoT Concept Support. Retrieved from https://ceur-ws.org/Vol-2923/paper30.pdf

81. Singh, S., Sharma, P. K., Moon, S. Y., & Park, J. H. (2017). Advanced Lightweight Encryption Algorithms for IoT Devices: Survey, Challenges and Solutions. Journal of Ambient Intelligence & Human Computing. https://doi.org/10.1007/s12652-017-0494-4.

82. Sokolov, V., Skladannyi, P., & Astapenya, V. (2023). Bluetooth Low-Energy Beacon Resistance to Jamming Attack. In 2023 IEEE 13th International

Conference on Electronics and Information Technologies (ELIT) (pp. 270-274). Lviv, Ukraine. DOI: 10.1109/ELIT61488.2023.10310815.

83. Sokolov, V., Skladannyi, P., & Hulak, H. (2022). Stability Verification of Self-Organized Wireless Networks with Block Encryption. CEUR Workshop Proceedings, 3137, 227-237. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-85130712118&partnerID=40&md5=93e600934ebfbb3454dc4817c1f767df

84. Sokolov, V., Skladannyi, P., & Korshun, N. (2023). ZigBee Network Resistance to Jamming Attacks. In 2023 IEEE International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo) (pp. 161-165). Kyiv, Ukraine. doi: 10.1109/UkrMiCo61577.2023.10380360.

85. Sokolov, V., Skladannyi, P., & Platonenko, A. (2023). Jump-Stay Jamming Attack on Wi-Fi Systems. In 2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT) (pp. 1-5). Lviv, Ukraine. doi: 10.1109/CSIT61576.2023.10324031.

86. Song, J., Lee, K., & Lee, H. (2013). Biclique cryptanalysis on lightweight block cipher: HIGHT and Piccolo. International Journal of Computer Mathematics, 90, 2564-2580. https://doi.org/10.1080/00207160.2013.767445

87. Sönmez Turan, M., McKay, K., Chang, D., Çalık, Ç., Bassham, L., Kang, J., & Kelsey, J. (2021). Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process. https://doi.org/10.6028/NIST.IR.8369.

88. South Korea Telecommunications Technology Associations (TTA). (2006). 64-bit Block Cipher HIGHT. Standardization Number TTAS.KO-12.0040. Retrieved from http://www.tta.or.kr/English/new/standardization/engttastddesc.jsp?stdno=TTAK.KO12.0040/R1.

89. Thakor, V. A., Razzaque, M. A., & Khandaker, M. R. A. (2021). Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A

Review, Comparison and Research Opportunities. IEEE Access, 9, 28177-28193. https://doi.org/10.1109/ACCESS.2021.3052867

90. Townsend, K., Cufí, C., Akiba, & Davidson, R. (2014). Getting Started with Bluetooth Low Energy. O'Reilly Media, Inc. ISBN: 9781491900581. Retrieved from https://www.oreilly.com/library/view/getting-started-with/9781491900550/

91. V. Sokolov, F. Kipchuk, P. Skladannyi, O. Zhyltsov, & D. Ageyev. (2022). Method for Increasing the Various Sources Data Consistency for IoT Sensors. In 2022 IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T) (pp. 522-526). Kharkiv, Ukraine. https://doi.org/10.1109/PICST57299.2022.10238518

92. Watanabe, Y., Yamamoto, H., & Yoshida, H. (2022). Performance Evaluation of NIST LWC Finalists on AVR ATmega and ARM Cortex-M3 Microcontrollers. Cryptology ePrint Archive, Paper 2022/1071. Retrieved from https://eprint.iacr.org/2022/1071

93. Wen, L., Wang, M., Bogdanov, A., & Chen, H. (2014). Multidimensional zero-correlation attacks on lightweight block cipher HIGHT: Improved cryptanalysis of an ISO standard. Information Processing Letters, 114(6), 322-330. https://doi.org/10.1016/j.ipl.2014.01.007

94. Wheeler, D., & Needham, R. (1997). TEA Extensions (Technical Report). Computer Laboratory, University of Cambridge, Cambridge. Retrieved from https://www.tayloredge.com/reference/Mathematics/TEA-XTEA.pdf

95. Xu, Y., Hao, Y., & Wang, M. (2023). Revisit two memoryless state-recovery cryptanalysis methods on A5/1. IET Information Security, 17. https://doi.org/10.1049/ise2.12120.

96. Yan, Z., Zhang, P., & Vasilakos, A. (2014). A survey on trust management for Internet of Things. Journal of Network and Computer Applications, 42, 120-134. https://doi.org/10.1016/j.jnca.2014.01.014.

97. Yang, G., Zhu, B., Suder, V., Aagaard, M. D., & Gong, G. (2015). The Simeck Family of Lightweight Block Ciphers. In International Workshop on

Cryptographic Hardware and Embedded Systems, Saint Malo, September 13-16, 2015 (pp. 307-329). https://doi.org/10.1007/978-3-662-48324-4_16.

98. Zhurakovskyi, B., Pliushch, O., Polishchuk, M., Korshun, N., & Obushnyi, S. (2023). CEUR Workshop Proceedings, Volume 3421, Pages 67-76, 2023 Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2023), Virtual, Online, 28 February 2023. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85163829603&partnerID=40&md5=3ee22afec77589c39513dd8ac38435d6

99. Адміністрація Державної служби спеціального зв'язку та захисту інформації України. (2007). Наказ № 141 від 20 липня 2007 року "Про затвердження Положення про порядок розроблення, виробництва та експлуатації засобів криптографічного захисту інформації". Retrieved from https://zakon.rada.gov.ua/laws/show/z0862-07#Text

100. Бессалов, А. В. (2017). Еліптичні криві в формі Едвардса і криптографія: монографія. Київ: ІВЦ "Видавництво "Політехника". Retrieved from: https://elibrary.kubg.edu.ua/id/eprint/21879/

101. Бурячок В. Л., Гулак Г.М., Складанний П. М. (2017). Швидкий алгоритм генерації підстановок багатоалфавітної заміни. Захист інформації, 2017(2), 173–177. https://doi.org/10.18372/2410-7840.19.11767

102. Горбенко, І. Д., & Горбенко, Ю. І. (2012). Прикладна криптологія. Теорія. Практика. Застосування: монографія. Харків: Форт. МОНМС України, Харк. нац. ун-т радіоелектроніки, ПАТ "Ін-т інформаційних технологій". ISBN 978-966-8599-99-6. Retrieved from https://catalogue.nure.ua/document=143671

103. Горбенко, І. Д., Гулак, Г. М., Олійников, Р. В., Руженцев, В. І., & Михайленко, М. С. (2005). Аналіз властивостей алгоритмів блокового симетричного шифрування (за результатами міжнародного проєкту NESSIE). В 8-ій Міжнародній науково-практичній конференції «Безпека інформації в інформаційно-телекомунікаційних системах». Тези доповідей (с. 17-18). Київ.

104. Гулак Г.М., Складанний П.М. (2017). Забезпечення гарантоздатності автоматизованих систем управління та передачі даних

безпілотних літальних апаратів. Математичні машини і системи, 2017(3), 154–161. Retrieved from http://www.immsp.kiev.ua/publications/articles/2017/2017_3/03_2017_Gulak.pdf

105. Гулак, Г., & Ковальчук, Л. (2001). Різні підходи до визначення випадкових послідовностей. Науково-технічний збірник "Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні", (вип. 3), 127-133. https://ela.kpi.ua/bitstream/123456789/15434/1/03_p127.pdf

106. Гулак, Г., Жданова, Ю., Складанний, П., Гулак, Є., & Корнієць, В. (2022). Уразливості шифрування коротких повідомлень в мобільних інформаційно-комунікаційних системах об'єктів критичної інфраструктури. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 1(17), 145–158. https://doi.org/10.28925/2663-4023.2022.17.145158

107. ДСТУ 9041:2020 Інформаційні технології. Криптографічний захист інформації. Алгоритм шифрування коротких повідомлень, заснованих на скручених еліптичних кривих Едвардса (2020). Retrieved from: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=90523

108. ДСТУ ISO/IEC 27002:2023 Інформаційна безпека, кібербезпека та захист конфіденційності. Засоби контролювання інформаційної безпеки (ISO/IEC 27002:2022, IDT) https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=104399

109. Звіт Cisco щорічного інтернет-аналізу за 2018–2023.(2020). Cisco. Взято 2022 з https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

110. Карпович, І. М., Гладка, О. М., & Наконечна, Ю. А. (2020). Аналіз ризиків безпеки інформаційної системи IT-підприємства. Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки, 31(70), 69–74. https://doi.org/10.32838/2663-5941/2020.5/12

111. Корнієць, В., & Черненко, Р. (2023). Модифікація криптографічного алгоритму a5/1 для забезпечення комунікацій пристроїв IoT.

Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 4(20), 253–271. https://doi.org/10.28925/2663-4023.2023.20.253271

112. Кузнецов, О. О., Олійников, Р. В., Горбенко, Ю. І., Пушкарьов, А. І., Дирда, О. В., & Горбенко, І. Д. (2014). Обгрунтування вимог, побудова та аналіз перспективних симетричних криптоперетворень на основі блочних шифрів. Вісник Національного університету "Львівська політехніка", (806), 124-140. Retrieved from https://science.lpnu.ua/sites/default/files/journal-paper/2017/nov/6634/21-124-141.pdf

113. Курбанмурадов, Д. М., Соколов, В. Ю., & Астапеня, В. М. (2019). Реалізація протоколу шифрування XTEA на базі безпровідових систем стандарту IEEE 802.15.4. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 2(6), 32–45. https://doi.org/10.28925/2663-4023.2019.6.3245

114. Лавренюк, С.І., Шелестов, А.Ю., & Лавренюк, А.М. (2010). Багатокритеріальний аналіз ризиків порушення безпеки інформації в Grid-системах. Проблеми програмування, (2-3), 507-512. Бібліографія: 16 назв. Retrieved from http://dspace.nbuv.gov.ua/handle/123456789/14667

115. Національний Банк України. (2011, 3 березня). Методичні рекомендації щодо впровадження системи управління інформаційною безпекою та методики оцінки ризиків відповідно до стандартів [Лист Національного Банку України від 03.03.2011 No 24-112/365]. https://zakon.rada.gov.ua/laws/show/v0365500-11#Text

116. НД ТЗІ 1.1-002-99. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу. (Затверджено наказом ДСТСЗІ СБ України від 28.04.99 р. № 22). Отримано з https://tzi.com.ua/downloads/1.1-002-99.pdf

117. НД ТЗІ 1.4-001-2000. Типове положення про службу захисту інформації в автоматизованій системі. (Затверджено наказом ДСТСЗІ СБ України від 04.12.2000 р. № 53). [Електронний ресурс]. Отримано з https://tzi.com.ua/downloads/1.4-001-2000.pdf

118. НД ТЗІ 3.7-003 -2005. Порядок проведення робіт із створення комплексної системи захисту інформації в інформаційно-телекомунікаційній системі. Київ. (Затверджено наказом ДСТСЗІ СБ України від 08.11.2005 р. №125). Отримано з https://tzi.com.ua/downloads/3.7-003-2005.pdf

119. Черненко, Р. (2023). Генерація псевдовипадкових послідовностей на мікроконтролерах з обмеженими обчислювальними ресурсами, джерела ентропії та тестування статистичних властивостей. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 2(22), 191–203. https://doi.org/10.28925/2663-4023.2023.22.191203.

120. Черненко, Р. (2023). Оцінка продуктивності алгоритмів легкої криптографії на обмежених 8-бітних пристроях. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 1(21). https://doi.org/10.28925/2663-4023.2023.21.273285

121. Черненко, Р. М., Рябчун, О. П., Ворохоб, М. В., Аносов, А. О., & Козачок, В. А. (2021). Підвищення рівня захищеності систем мережі інтернету речей за рахунок шифрування даних на пристроях з обмеженими обчислювальними ресурсами. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 3(11), 124–135. https://doi.org/10.28925/2663-4023.2021.11.124135.

ДОДАТОК А

Акти та довідки впровадження результатів дисертаційного дослідження

КИЇВСЬКИЙ СТОЛИЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ БОРИСА ГРІНЧЕНКА

ФАКУЛЬТЕТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ТА МАТЕМАТИКИ
вул. Левка Лук'яненка, 13-Б, м. Київ, Україна, 04207
Тел.: +380 44 428-34-14
fitm.kubg.edu.ua, fitm@kubg.edu.ua

BORYS GRINCHENKO
KYIV METROPOLITAN UNIVERSITY

FACULTY
OF INFORMATION TECHNOLOGIES
AND MATHEMATICS
13-B Levka Lukianenka St, Kyiv, Ukraine, 04207
Tel.: +380 44 428-34-14
fitm.kubg.edu.ua, fitm@kubg.edu.ua

15.01.2024 № 1

АКТ
про впровадження результатів дисертаційного дослідження
Черненка Романа Миколайовича
на тему «Моделі та методи забезпечення захисту інформації, що передається відкритими каналами в мережах інтернету речей»,
поданої на здобуття наукового ступеня доктора філософії
зі спеціальності 125 Кібербезпека та захист інформації

Цим Актом, грунтуючись на рішенні кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка Факультету інформаційних технологій та математики Київського столичного університету імені Бориса Грінченка, засвідчуємо, що нижчеперелічені наукові положення, а саме:

- вперше запропонований метод криптографічного захисту інформації в мережі інтернету речей на основі модифікованого алгоритму A5/1, що забезпечує підвищену стійкість шифрування та імітостійкість завдяки застосуванню байтової обробки інформації та застосування вузла накладання шифру на основі змінного латинського квадрату. Алгоритм має високу швидкодію, яка на 30,70 % більше ніж у відомих алгоритмів для інтернету речей;

- вдосконалений стандартний протокол Shockburst безпроводового інформаційного обміну в мережі, з метою безпечного формування сеансових ключів та забезпечення криптографічно захищеної передачі даних від пристроїв з обмеженими обчислювальними ресурсами до шлюзу;

- подальшого розвитку набула модель загроз для побудови системи захисту інформації, що обробляється пристроями з обмеженими обчислювальними ресурсами в мережі інтернету речей.

Розроблені особисто Черненком Романом Миколайовичем у ході проведення ним дисертаційних досліджень та отримали високу оцінку при обговоренні на засіданнях кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка Факультету інформаційних технологій та математики Київського столичного університету імені Бориса Грінченка.

Зазначені наукові результати:

по-перше, впроваджені в освітній процес кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка Факультету інформаційних технологій та математики Київського столичного університету імені Бориса Грінченка у робочих програмах навчальних дисциплін спеціальності 125 Кібербезпека за захист інформації першого (бакалаврського), другого (магістерського) та третього (освітньо-наукового) рівнів вищої освіти;

по-друге, впроваджені в програмно-апаратне забезпечення лабораторій безпеки інформаційних активів, антивірусного захисту інформації, систем технічного та криптографічного захисту інформації.

Дослідження Черненка Романа Миколайовича відповідає всім вимогам до організації наукового пошуку та дає позитивний результат у практичному застосуванні.

Декан
Факультету інформаційних технологій та математики
кандидат фізико-математичних наук,
старший науковий співробітник Оксана ЛИТВИН

ТОВ «Технологічні ІТ рішення»
03061, м. Київ, вул. Шепелєва Миколи, буд. 6
IBAN UA683808380000026008700446689
в АТ «ПРАВЕКС БАНК», Код МФО 380838
ЄДРПОУ 42116114 ІПН 421161126585
Тел. +380964600502 email: office@tit.solutions

№ 1 від 16.01.2024

Про впровадження результатів
дисертаційного дослідження

ДОВІДКА

**про впровадження результатів дисертаційного дослідження
Черненка Романа Миколайовича
на тему: «Моделі та методи забезпечення захисту інформації, що передається відкритими каналами в мережах інтернету речей»**

Цією довідкою засвідчено, що наукові результати дисертаційного дослідження на здобуття наукового ступеня доктора філософії за спеціальністю 125 Кібербезпека та захист інформації, Черненка Романа Миколайовича були використані в роботі ТОВ «Технологічні ІТ рішення».

Для підприємства представляє цінність та заслуговують на увагу запропонований метод криптографічного захисту інформації в мережі інтернету речей на основі модифікованого алгоритму A5/1, що забезпечує підвищену стійкість шифрування та імітостійкість завдяки застосування байтової обробки інформації та застосування вузла накладання шифру на основі змінного латинського квадрату. Алгоритм забезпечує високу швидкодію, яка на 30,70 % більше ніж у відомих алгоритмів для інтернету речей;

Вдосконалений стандартний протокол Shockburst безпроводового інформаційного обміну в мережі з метою безпечного формування сеансових ключів та забезпечення криптографічного захисту передачі даних від пристроїв з обмеженими обчислювальними ресурсами до шлюзу.

Результати дослідження мають практичне застосування для криптографічного захисту інформації, що передається пристроями з обмеженими обчислювальними ресурсами.

Директор
ТОВ «Технологічні ІТ рішення» Сергій БАРСУЧЕНКО

2d:3d

Web: https://odo.com.ua
E-mail: info@odo.com.ua
Address: office 123, 12, Anton Tsedik str., Kyiv, 03057, Ukraine

16.01.2024 №____

Про впровадження результатів
дисертаційного дослідження

ДОВІДКА

**про впровадження результатів
отриманих при виконанні дисертаційного дослідження
Черненка Романа Миколайовича
на тему: «Моделі та методи забезпечення захисту інформації, що передається відкритими каналами в мережах інтернету речей»**

Ця довідка свідчить про те, що результати, отримані в дисертаційному дослідженні Черненка Романа Миколайовича, були використані в ході роботи ТОВ «2Д3Д» щодо підвищення рівня безпеки інформації, що передається пристроями з обмеженими обчислювальними ресурсами.

Перелік реалізації та впровадження результатів дослідження:

– вперше запропоновано метод криптографічного захисту інформації в мережі інтернету речей на основі модифікованого алгоритму A5/1, що забезпечує підвищену стійкість шифрування та імітостійкість завдяки застосуванню байтової обробки інформації та застосування вузла накладання шифру на основі змінного латинського квадрату. Алгоритм має високу швидкодію, яка на 30,70 % більше ніж у відомих алгоритмів для інтернету речей;

– вдосконалено стандартний протокол Shockburst безпроводового інформаційного обміну в мережі, з метою безпечного формування сеансових ключів та забезпечення криптографічно захищеної передачі даних від пристроїв з обмеженими обчислювальними ресурсами до шлюзу.

Наукові та практичні результати дослідження можуть знайти подальше практичне застосування в процесі розробки та удосконалення існуючих механізмів забезпечення кібербезпеки мереж інтернету речей, що побудовані на основі пристроїв з обмеженими обчислювальними ресурсами.

Директор
ТОВ «2Д3Д» **Артем ПАЛАДЮК**

"2Д3Д" Ідентифікаційний код 41008488

ДОДАТОК Б

Лістинг програми керування рухом регістрів та формування шифруючої послідовності

```
byte reg1[19];
byte reg1_13 = 13;
byte reg1_16 = 16;
byte reg1_17 = 17;
byte reg1_18 = 18;
byte reg1Clock = 8;
byte reg2[22];
byte reg2_20 = 20;
byte reg2_21 = 21;
byte reg2Clock = 10;
byte reg3[23];
byte reg3_7 = 7;
byte reg3_20 = 20;
byte reg3_21 = 21;
byte reg3_22 = 22;
byte reg3Clock = 10;
byte X[256];
void shiftReg1(void) {
   reg1[reg1_18] = (reg1[reg1_18] + reg1[reg1_17] + reg1[reg1_16] +
reg1[reg1_13])%256;
   if (--reg1_18 == 255) reg1_18 = 18;
   if (--reg1_13 == 255) reg1_13 = 18;
   if (--reg1_16 == 255) reg1_16 = 18;
   if (--reg1_17 == 255) reg1_17 = 18;
   if (--reg1Clock == 255) reg1Clock = 18;
}
void shiftReg2(void) {
   reg2[reg2_21] = (reg2[reg2_21] + reg2[reg2_20])%256;
   if (--reg2_21 == 255) reg2_21 = 21;
   if (--reg2_20 == 255) reg2_20 = 21;
   if (--reg2Clock == 255) reg2Clock = 21;
}
void shiftReg3(void) {
   reg3[reg3_22] = (reg3[reg3_22] +
reg3[reg3_21]+reg3[reg3_20]+reg3[reg3_7])%256;
   if (--reg3_22 == 255) reg3_22 = 22;
   if (--reg3_21 == 255) reg3_21 = 22;
   if (--reg3_20 == 255) reg3_20 = 22;
   if (--reg3_7 == 255) reg3_7 = 22;
   if (--reg3Clock == 255) reg3Clock = 22;
}
void majorityFunctionAndShift(){
    byte x = (byte)(reg1[reg1Clock] & 1);
    byte y = (byte)(reg2[reg2Clock] & 1);
    byte z = (byte)(reg3[reg3Clock] & 1);
    byte f = (byte)(x & y | x & z | y & z);
    if (x == f) {shiftReg1(); }
    if (y == f) {shiftReg2(); }
    if (z == f) {shiftReg3(); }
}
byte getKeyByte() {
  return (reg1[18] + reg2[reg2_21] + reg3[reg3_22])%256;
}
byte encA5_128(byte M){
    byte C = X[(getKeyByte()+M)%256];
    return C;}
```

ДОДАТОК В

Список опублікованих праць за темою дисертації

1. Черненко, Р. М., Рябчун, О. П., Ворохоб, М. В., Аносов, А. О., & Козачок, В. А. (2021). Підвищення рівня захищеності систем мережі інтернету речей за рахунок шифрування даних на пристроях з обмеженими обчислювальними ресурсами. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 3(11), 124–135. https://doi.org/10.28925/2663-4023.2021.11.124135

2. Chernenko, R., Anosov, A., Kyrychok, R., Brzhevska, Z., & Spasiteleva, S. (2022). Encryption Method for Systems with Limited Computing Resources. CEUR Workshop Proceedings, 3288, pp. 142-148. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85143827975&partnerID=40&md5=fc5b960373acefb92e4755f0a571afb2

3. Корнієць, В., & Черненко, Р. (2023). Модифікація криптографічного алгоритму a5/1 для забезпечення комунікацій пристроїв IoT. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 4(20), 253–271. https://doi.org/10.28925/2663-4023.2023.20.253271

4. Черненко, Р. (2023). Оцінка продуктивності алгоритмів легкої криптографії на обмежених 8-бітних пристроях. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 1(21). https://doi.org/10.28925/2663-4023.2023.21.273285

5. Черненко, Р. (2023). Генерація псевдовипадкових послідовностей на мікроконтролерах з обмеженими обчислювальними ресурсами, джерела ентропії та тестування статистичних властивостей. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 2(22), 191–203. https://doi.org/10.28925/2663-4023.2023.22.191203