Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня магістра

студента *Шульгіна Олексія Леонідовича*

академічної групи *125м-21з-1*

спеціальності *125 Кібербезпека*

спеціалізації

за освітньо-професійною програмою *Кібербезпека*

На тему *Забезпечення кібербезпеки пристроїв інтернету речей на базі мікроконтролера ESP8266*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
Кваліфікаційної роботи	проф. Кагадій Т.С.			
розділів:				
спеціальний	ас. Мілінчук Ю.А.			
економічний	к.е.н., доц. Пілова Д.П.			
Рецензент				
Нормоконтролер	ст. викл. Тимофєєв Д.С.			

Дніпро
2022

ЗАТВЕРДЖЕНО:
завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.
«___»_____2022 року

ЗАВДАННЯ

на кваліфікаційну роботу ступеня магістра

студенту *Шульгіну О.Л.* **академічної групи** *125м-21з-1*
(прізвище та ініціали) (шифр)

спеціальності *125 Кібербезпека*

спеціалізації _____

за освітньо-професійною програмою *Кібербезпека*

На тему *Забезпечення кібербезпеки пристроїв інтернету речей на базі мікроконтролера ESP8266*

Затверджену наказом ректора НТУ «Дніпровська політехніка» від _____ № _____

Розділ	Зміст	Термін виконання
Стан питання. Постановка задачі	Дослідження загроз та вразливостей пристроїв інтернету речей на базі контролера ESP8266	
Спеціальна частина	Забезпечення кібербезпеки пристроїв інтернету речей на базі контролера ESP8266	
Економічна частина	Розрахунок економічної складової	

Завдання видано _____ *Мілінчук Ю.А.*
(підпис керівника) (прізвище та ініціали)

Дата видачі завдання: _____

Дата подання до екзаменаційної комісії _____

Прийнято до виконання _____ *Шульгін О.Л.*
(підпис студента) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка: 81 с., 2 рис., 5 табл., 4 додатки, 41 джерело.

Об'єкт дослідження: пристрої інтернету речей на базі контролера ESP8266.

Предмет дослідження: кіберзахист пристроїв інтернету речей на базі контролера ESP8266.

Мета роботи: аналіз існуючих вразливостей та загроз для пристроїв інтернету речей на базі контролера ESP8266 і розробка методів підвищення їх кібербезпеки.

Методи розробки: порівняння, аналіз, опис.

В першому розділі надано загальну характеристику пристроїв інтернету речей і контролера ESP8266, розглянуто його застосування в автоматизованих системах керування та системах інтернету речей. Проаналізовано вразливості та загрози, пов'язані з пристроями інтернету речей на базі контролера ESP8266, розглянуто існуючі методи кіберзахисту.

У спеціальній частині запропоновано способи забезпечення кібербезпеки пристроїв інтернету речей, побудованих з використанням контролера ESP8266.

В економічному розділі визначено економічну доцільність забезпечення кіберзахисту систем інтернету речей. Проведено розрахунок капітальних (фіксованих) витрат, поточних (експлуатаційних) витрат, загального збитку від атаки на ІТС та загального ефекту від впровадження рекомендацій.

Наукова новизна роботи полягає у застосуванні методів кіберзахисту до пристроїв, що використовують контролер ESP8266.

Практичне значення роботи полягає у підвищенні кіберзахисту автоматизованих систем керування, в тому числі пристроїв інтернету речей, побудованих на базі контролера ESP8266 або з його застосуванням.

Ключові слова: КІБЕРБЕЗПЕКА, АВТОМАТИЗОВАНА СИСТЕМА КЕРУВАННЯ, ІНТЕРНЕТ РЕЧЕЙ, РОЗУМНИЙ ДІМ, WI-FI, ESP8266

THE ABSTRACT

Explanatory note: 81 pp., 2 fig., 5 tab., 4 additions, 41 sources.

The object of research: Internet of Things devices based on the ESP8266 controller.

The subject of research: cyber protection of Internet of Things devices based on the ESP8266 controller.

The purpose of the work: analysis of existing vulnerabilities and threats for Internet of Things devices based on the ESP8266 controller and development of methods to increase their cyber security.

Development methods: comparison, analysis, description.

In the first section, the general characteristics of Internet of Things devices and the ESP8266 controller are given, and its application in automated control systems and Internet of Things systems is considered. Vulnerabilities and threats related to Internet of Things devices based on the ESP8266 controller were analyzed, and existing cyber protection methods were considered.

The special part offers ways to ensure cyber security of Internet of Things devices built using the ESP8266 controller.

The economic section defines the economic feasibility of ensuring cyber protection of Internet of Things systems. The calculation of capital (fixed) costs, current (operating) costs, total damage from an attack on ITS and the total effect from the implementation of the recommendations was carried out.

The scientific novelty of the work consists in the application of cyber protection methods to devices using the ESP8266 controller.

The practical significance of the work consists in increasing the cyber protection of automated control systems, including Internet of Things devices built on the basis of the ESP8266 controller or with its use.

Keywords: CYBER SECURITY, AUTOMATED CONTROL SYSTEM, INTERNET OF THINGS, SMART HOME, WI-FI, ESP8266

СПИСОК УМОВНИХ СКОРОЧЕНЬ

AES – Advanced Encryption Standard

API – Application Programming Interfaces

CCMP – Counter Mode with Cipher Block Chaining Message Authentication Code Protocol

CPU – Central Processing Unit

DoS – Denial of Service

DDoS – Distributed Denial of Service

EAP – Extensible Authentication Protocol

FOTA – Firmware Over the Air

GPS – Global Positioning System

HTTP – HyperText Transfer Protocol

HTTPS – HyperText Transfer Protocol Secure

IDE – Integrated Development Environment

IEEE – Institute of Electrical and Electronics Engineers

I^2C – Inter-Integrated Circuit

I^2S – Inter-IC Sound

IIoT – Industrial Internet of Things

IoBT – Internet of Battlefield Things

IoMT – Internet of Military Things

IoT – Internet of Things

IP – Internet Protocol

LAN – Local Area Network

MAC – Media Access Control

OTA – Over the Air

PLC – Power Line Communication

RFID – Radio Frequency Identification

RSN – Robust Security Network

RTOS – Real-Time Operating System

SDIO – Secure Digital Input Output

SDK – Software Development Kit

SoC – System-on-a-Chip

SOTA – Software Over the Air

SPI – Serial Peripheral Interface

SSID – Service Set Identifier

TCP – Transmission Control Protocol

TKIP – Temporal Key Integrity Protocol

UART – Universal Asynchronous Receiver/Transmitter

UDP – User Datagram Protocol

WEP – Wired Equivalent Privacy

WLAN – Wireless Local Area Network

WPA – Wi-Fi Protected Access

АСК – автоматизована система керування

АСКТП – автоматизована система керування технологічним процесом

АЦП – аналого-цифровий перетворювач

БД – база даних

ІБ – інформаційна безпека

МПК – мікропроцесорний контролер

ОЗП – оперативний запам’ятовуючий пристрій

ОС – операційна система

ПЗ – програмне забезпечення

ПЗП – постійний запам’ятовуючий пристрій

ПК – персональний комп’ютер

ПЛК – програмований логічний контролер

СКБД – система керування базами даних

ШІ – штучний інтелект

ШІМ – широтно-імпульсна модуляція

ЗМІСТ

ВСТУП

Сучасний світ як ніколи оснащений великою кількістю засобів зв’язку. Світову економіку формують люди, які спілкуються, перебуваючи в різних часових поясах, і отримують доступ до важливої інформації звідусіль. Кібербезпека стимулює продуктивність і впровадження інновацій, що дає користувачам змогу впевнено працювати та спілкуватись онлайн. Правильні рішення й процеси дають змогу компаніям і державним установам користуватися технологіями для покращення спілкування та надання послуг, не ризикуючи постраждати від атак [1].

Інтернет речей (IoT) поєднує повсякденні «речі» з Інтернетом. Це поняття відноситься до колективної мережі підключених пристроїв та технології, що полегшує зв'язок між пристроями та хмарою, а також між самими пристроями. Завдяки появі недорогих комп'ютерних мікросхем та телекомунікацій з високою пропускною спроможністю зараз існують мільярди пристроїв, що мають підключення до Інтернету. Це означає, що повсякденні пристрої, такі як зубні щітки, пилососи, автомобілі та механічні установки можуть використовувати датчики для збору даних та розумного реагування на дії користувачів [2].

Останнім часом вартість інтеграції обчислювальної потужності до невеликих об'єктів значно знизилася. Виникла ціла індустрія, спрямована на наповнення будинків, підприємств та офісів пристроями IoT. Ці смарт-об'єкти можуть автоматично передавати дані до Інтернету та приймати з Інтернету [2].

Оскільки все більше сучасних пристроїв та систем безпеки підключаються до мережі, кібербезпека Інтернету речей стає все більш серйозною проблемою. IoT з'єднує мільярди пристроїв з Інтернетом, і це також розширює потенціал кібератак проти мереж та інформації. Захист пристроїв інтернету речей матиме вирішальне значення для розширення інтернету речей та підвищення його безпеки.

1 СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ

1.1 Інтернет речей

Інтернет речей (Internet of Things, IoT) – це концепція мережі, яка складається із взаємозв'язаних фізичних пристроїв, що мають вбудовані датчики, а також програмне забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами в автоматичному режимі, за допомогою використання стандартних протоколів зв'язку. Окрім датчиків, мережа може мати виконавчі пристрої, вбудовані у фізичні об'єкти і пов'язані між собою через дротові чи бездротові мережі. Ці взаємопов'язані пристрої мають можливість зчитування та приведення в дію, функцію програмування та ідентифікації, а також дозволяють виключити необхідність участі людини, за рахунок використання інтелектуальних інтерфейсів [3].

Концепція інтернету речей була сформульована як осмислення перспектив широкого застосування засобів радіочастотної ідентифікації для взаємодії фізичних предметів між собою та із зовнішнім оточенням. Наповнення концепції різноманітним технологічним змістом та впровадження практичних рішень для її реалізації починаючи з 2010-х років вважається стійкою тенденцією в інформаційних технологіях, насамперед завдяки повсюдному поширенню бездротових мереж, появі хмарних обчислень, розвитку технологій міжмашинної взаємодії, початку активного переходу на IPv6 програмно-визначуваних мереж.

Основною концепцією інтернету речей є можливість підключення різних об'єктів (речей), які людина може використовувати в повсякденному житті, наприклад, холодильник, кондиціонер, автомобіль, велосипед і навіть кросівки. Всі ці об'єкти (речі) повинні бути оснащені вбудованими давачами або сенсорами, які мають можливість обробляти інформацію, що надходить з навколишнього середовища, обмінюватися нею і виконувати різні дії в залежності від отриманої інформації. Прикладом впровадження такої концепції є система «розумний будинок» або «розумна ферма». Ця система аналізує дані навколишнього середовища і в залежності від показників регулює температуру в приміщенні. У

зимовий період регулюються інтенсивність опалення, а в разі спекотної погоди будинок має механізми відкривання і закривання вікон, завдяки чому провітрюється будинок, і все це відбувається без втручання людини [3].

Термін «Інтернет речей» вперше був введений Кевіном Ештоном у 1999 року під час його роботи над Procter & Gamble, щоб описати систему, в якій фізичні об'єкти могли бути пов'язані з давачами і мережею Інтернет. Ештон ввів цей термін, щоб проілюструвати можливості радіочастотної ідентифікації (RFID), яка використовується в корпоративних системах поставок, щоб порахувати і відстежити товари без потреби в людському втручанні [3].

RFID – це спосіб автоматичної ідентифікації об'єктів, у якому за допомогою радіосигналів зчитуються або записуються дані, що зберігаються в так званих транспондерах, або RFID-мітках. Будь-яка RFID-система складається зі зчитувача (зчитувач) і транспондера (RFID-мітка). Більшість RFID-міток складається з двох частин. Перша частина – інтегральна схема для зберігання та обробки інформації, модулювання і демодулювання радіочастотного сигналу та деяких інших функцій. Друга частина – антена для прийому та передачі сигналу [4].

У 2004 році в Scientific American було опубліковано велику статтю [5], яка була присвячена «інтернету речей» і наочно демонструвала можливості концепції для побутового застосування. У статті наведена ілюстрація, на якій показано, як побутові прилади (будильник, кондиціонер), домашні системи (система садового поливу, охоронна система, система освітлення), датчики (теплові, датчики освітленості та руху) та «речі» (наприклад, лікарські препарати, забезпечені ідентифікаційною міткою) взаємодіють один з одним за допомогою комунікаційних мереж (інфрачервоних, бездротових, силових та слаботочних мереж) та забезпечують повністю автоматичне виконання процесів (включають кавоварку, змінюють освітленість, нагадують про прийом ліків, підтримують температуру, забезпечують полив саду, дозволяють зберігати енергію та керувати її споживанням). Самі по собі представлені варіанти домашньої автоматизації не були новими, але наголос у публікації на об'єднанні пристроїв і «речей» в єдину обчислювальну мережу, що обслуговується інтернет-протоколами, і розгляд

«інтернету речей» як особливого явища сприяли набуттю концепції широкої популярності [6].

У звіті Національної розвідувальної ради США 2008 року «інтернет речей» фігурує як одна з шести підривних технологій, зазначається, що повсюдне та непомітне для споживачів перетворення на інтернет-вузли таких поширених речей, як товарна упаковка, меблі, паперові документи, може помітно підвищити ризики у сфері національної інформаційної безпеки [6].

Період з 2008 по 2009 рік аналітики корпорації Cisco вважають «справжнім народженням інтернету речей», оскільки, за їх оцінками, саме в цьому проміжку кількість пристроїв, підключених до глобальної мережі, перевищила чисельність населення Землі, тим самим «інтернет людей» став «інтернетом речей» [7].

З 2009 року за підтримки Єврокомісії в Брюсселі щорічно проводиться конференція «Internet of Things», на якій представляють доповіді єврокомісари та депутати Європарламенту, урядовці з європейських країн, керівники таких компаній, як SAP, SAS Institute, Telefónica, провідні вчені великих університетів та дослідницьких лабораторії [6].

З початку 2010-х років «інтернет речей» стає рушійною силою парадигми «туманних обчислень», що поширює принципи хмарних обчислень від центрів обробки даних до величезної кількості географічно розподілених пристроїв, що взаємодіють, яка розглядається як платформа «інтернету речей» [6].

Типова система інтернету речей працює за допомогою збору та обміну даними в режимі реального часу. Система інтернету речей складається з наступних чотирьох компонентів [8]:

- Пристрої (речі). Це практично будь-які об’єкти, які можливо уявити. Починаючи від спеціалізованих датчиків, мікроконтролерів і виконавчих механізмів, до повсякденних предметів, такі як звичайні кавоварки чи розумні автомобілі, і закінчуючи такими незвичайними предметами, як рослини, все сьогодні можна підключити до Інтернету і таким чином зробити об’єкт частиною інтернету речей. Завдання розумних речей включають не тільки простий збір даних, але, перш за все, взаємодію один з одним і віддаленим сервером або хмарою

(залежно від використовуваної технології) і виконання дій на основі отриманих команд.

– Підключення. Якщо підключені пристрої призначені для того, щоб обмінюватись даними про те, що вони «відчувають» або де вони знаходяться, їм потрібно надати мову, якою вони можуть це робити, і канал, через який вони можуть передавати свої «висловлювання». Все це забезпечується компонентом підключення IoT. Тоді як «мова» представлена конкретними протоколами Інтернету речей, які використовуються в певному розгортанні, канал зв'язку забезпечується добре відомими рішеннями підключення, такими як Wi-Fi або Bluetooth, або менш відомими технологіями, включаючи Long Range Wide Area. Мережі (з провідними стандартами, такими як Sigfox або Narrowband IoT). Застосування даного рішення багато в чому залежить від вимог і обмежень конкретного IoT-проекту.

– Програмне забезпечення. Це та частина IoT, яка зазвичай прихована від очей звичайного кінцевого користувача IoT, але насправді це справжнє поле діяльності IoT. Програмне забезпечення – це те, що дає речам здатність «думати» та діяти. Завдяки цьому дані, зібрані розумними об'єктами, можна структурувати, аналізувати та керувати ними. Виходячи з цього, програмне забезпечення вирішує, чи потрібно вживати будь-яких дій або потрібно сповістити користувача.

– Додатки. Це та частина, де кінцевий користувач вступає в дію та бере на себе контроль. Завдяки прикладному рівню зібрані та проаналізовані дані візуалізуються, а користувачеві надається корисна інформація про роботу всієї системи. Більше того, він може впливати на спосіб керування пристроями та отримує попередження, коли виникає потреба в діях людини, що зазвичай трапляється, коли можливості самоконтролю системи вичерпані.

Згідно з матеріалами [2] AWS, сукупність пристроїв інтернету речей може розглядатися, залежно від масштабів та структури, в якості окремих мереж, таких як «розумний дім» або «розумна будівля» та «розумне місто».

Пристрої «розумного дому» в основному орієнтовані на підвищення ефективності та безпеки будинку, а також на поліпшення домашніх мереж. Такі

пристрої, як «розумні» розетки, контролюють споживання електроенергії, а інтелектуальні термостати забезпечують підвищений контроль температури. Гідропонні системи можуть використовувати датчики інтернету речей для управління садом, а датчики диму можуть виявляти тютюновий дим. Домашні системи безпеки, такі як дверні замки, камери відеоспостереження та детектори витоку води, можуть виявляти та запобігати загрозам, а також посилати попередження власникам будинків [2].

В системах «розумного дому» пристрої застосовуються для [2]:

- автоматичного виключення невикористовуваних пристроїв;
- управління орендованою нерухомістю та її обслуговування;
- пошук предметів, які знаходяться в такому місці, як ключі або гаманці.
- автоматизації щоденних завдань, таких як прибирання, приготування їжі тощо.

«Розумні будівлі», якими можуть бути, наприклад, комерційні будівлі, використовують програми інтернету речей для підвищення операційної ефективності [2].

Пристрої IoT можуть використовуватися в «розумних будівлях» для [2]:

- скорочення витрат електроенергії;
- зменшення витрат на обслуговування;
- Найефективнішого використання робочого простору.

В «розумних містах» застосування інтернету речей зробило міське планування та обслуговування інфраструктури більш ефективними. Уряди використовують програми IoT для вирішення проблем в інфраструктурі, охороні здоров'я та навколишньому середовищі [2].

Інтернет речей в «розумних містах» використовується для [2]:

- оцінювання якості повітря та рівня радіації;
- скорочення рахунків за електроенергію за допомогою «розумних» систем висвітлення.
- виявлення потреб в обслуговуванні критично важливих інфраструктур, таких як вулиці, мости та трубопроводи;

- збільшення прибутку за рахунок ефективного керування паркуванням.

У статті [9], що опублікована IEEE, запропоновано класифікувати пристрої інтернету речей за наступними категоріями:

- розумні пристрої, що носяться
- розумний дім
- розумне місто
- розумне середовище
- розумне підприємство

Розглянемо детальніше найбільш розповсюджені можливості застосування пристроїв інтернету речей у побуті. В системах розумного дому пристрої інтернету речей є частиною ширшої концепції домашньої автоматизації, яка може включати освітлення, опалення та кондиціювання повітря, медіа-системи та системи безпеки, а також системи відеоспостереження. Це дозволяє підвищити економію енергії за рахунок автоматичного відключення світла та електроніки або за рахунок інформування мешканців будинку про використання. Одним з ключових застосувань інтернету речей є надання допомоги людям з обмеженими можливостями та людям похилого віку. Такі системи використовують допоміжні технології для особливих потреб власника. Пристрої інтернету речей можна використовувати для забезпечення віддаленого моніторингу стану здоров'я та систем оповіщення про надзвичайні ситуації [6].

Промисловий інтернет речей, також відомий як IIoT, отримує та аналізує дані від підключеного обладнання, операційних технологій, місць розташування та людей. У поєднанні з пристроями моніторингу операційних технологій IIoT допомагає регулювати та контролювати промислові системи. Інтернет речей дозволяє також підключати різні виробничі пристрої, оснащені функціями виявлення, ідентифікації, обробки, зв'язку, приведення в дію та створення мереж. Мережевий контроль та управління виробничим обладнанням, управління активами та ситуаціями або управління виробничими процесами дозволяють використовувати IoT для промислових програм та інтелектуального виробництва. Інтелектуальні системи інтернету речей дозволяють швидко виробляти та

оптимізувати нові продукти, а також швидко реагувати на потреби у продуктах. Цифрові системи управління для автоматизації управління технологічними процесами, інструменти оператора та системи службової інформації для оптимізації безпеки та охорони обладнання входять до компетенції IIoT. Існує безліч додатків інтернету речей у сільському господарстві, таких як збір даних про температуру, кількість опадів, вологість, швидкість вітру, зараженість шкідниками та склад ґрунту. Ці дані можуть бути використані для автоматизації методів ведення сільського господарства, прийняття обґрунтованих рішень щодо покращення якості та кількості, мінімізації ризиків та відходів, а також для скорочення зусиль, необхідних для керування посівами [6].

Інтернет речей може бути застосований для моніторингу та контролю функціонування стійкої міської та сільської інфраструктури, такої як мости, залізничні колії, вітряні електростанції на суші та в морі тощо. Інфраструктурний інтернет речей може використовуватися для моніторингу будь-яких подій або змін у структурних умовах, які можуть загрожувати безпеці та збільшити ризик. Інтернет речей може допомогти в інтеграції комунікацій, управління та обробки інформації у різних транспортних системах. Застосування Інтернету поширюється на всі аспекти транспортних систем. Динамічна взаємодія між цими компонентами транспортної системи забезпечує зв'язок між транспортними засобами та всередині них, інтелектуальне керування рухом, інтелектуальне паркування, електронні системи стягування плати, логістику та керування автопарком, керування транспортними засобами, безпеку та допомогу на дорогах. Значна кількість енергоспоживаючих пристроїв (наприклад, лампи, побутова техніка, двигуни, насоси тощо) вже інтегрують підключення до Інтернету, що дозволяє їм взаємодіяти з комунальними службами не лише для балансування вироблення електроенергії, а й допомагає оптимізувати споживання енергії загалом. Додатки IoT для екологічного моніторингу зазвичай використовують датчики для сприяння охороні навколишнього середовища шляхом моніторингу якості повітря або води, атмосферних або ґрунтових умов і можуть навіть включати такі області, як моніторинг переміщень диких тварин та їх місць проживання. Іншим прикладом

інтеграції Інтернету речей є жива лабораторія, яка поєднує та поєднує дослідницькі та інноваційні процеси, створюючи в рамках державно-приватного партнерства людей [6].

Інтернет речей також може бути застосований у військовій галузі. Інтернет військових речей (IoMT) – це застосування технологій IoT у військовій галузі для розвідки, спостереження та інших цілей, пов'язаних з бойовими діями. Такі системи передбачають використання датчиків, боєприпасів, транспортних засобів, роботів, біометричних даних, придатних для носіння людиною та інших інтелектуальних технологій, які актуальні на полі бою [6]. Це складна мережа взаємопов'язаних об'єктів, або «речей», у військовій сфері, які постійно обмінюються даними один з одним, щоб координувати, навчатися та взаємодіяти з фізичним середовищем для виконання широкого спектру дій більш ефективним та поінформованим способом. Концепція IoMT значною мірою керується ідеєю про те, що в майбутніх військових битвах домінуватиме машинний інтелект і кібервійна, і вони, ймовірно, відбуватимуться в міському середовищі. Створюючи мініатюрну екосистему інтелектуальних технологій, здатних дистилювати сенсорну інформацію та автономно керувати декількома завданнями одночасно, IoMT концептуально розроблений, щоб звільнити більшу частину фізичного та психічного тягаря, з яким стикаються бійці під час бойових дій [10]. Інтернет речей на полі бою (IoBT) – це проект, ініційований та виконуваний Дослідницькою лабораторією армії США, який фокусується на фундаментальних науках, пов'язаних із IoT, що розширюють можливості солдатів армії [6]. IoBT з'єднує солдатів із розумними технологіями в броні, радіостанціях, зброї та інших об'єктах, щоб скоротити затримку циклів прийняття рішень, підвищити стійкість тактичної аналітики на полі бою та підштовхнути спеціалізований машинний інтелект до точки потреби [11]. Оскільки середовище IoBT дуже неоднорідне з точки зору пристроїв, мережевих стандартів, платформ, зв'язку тощо, виникають проблеми довіри, безпеки та конфіденційності, коли суб'єкти поля бою обмінюються інформацією один з одним [12].

Щоб об'єднати повсякденні речі у мережу інтернету речей необхідно застосувати декілька технологій.

- Для ідентифікації пристроїв можливе застосування мікросхем RFID, які мають свій індивідуальний номер та здатні без власного джерела живлення передавати інформацію приладам зчитування, або можливе використання QR-кодів, що скануються камерою смартфону. Для визначення точного місця знаходження застосовується технологія GPS, що ефективно використовується у смартфонах та навігаторах.
- Для відслідковування змін у стані елементу чи оточуючого середовища пристрої повинні оснащуватися сенсорами.
- Для обробки та накопичення даних з сенсорів повинен використовуватися вбудований комп'ютер або мікропроцесорний контролер.
- Для обміну інформацією між пристроями можуть бути використані технології бездротових мереж (Wi-Fi, Bluetooth, ZigBee, 6LoWPAN).
- Для передачі даних використовуються оптимізовані та «легкі» протоколи типу MQTT. Вони ґрунтуються на принципах публікації і підписок де кожен пристрій (давач чи сенсор) взаємодіє з програмою на сервері (брокером).

Взаємодія з Інтернетом передбачає, що пристрої мають використовувати IP-адресу в якості унікального ідентифікатора. Через обмежені адресні простори в IPv4, пристроям інтернету речей доведеться використовувати IPv6, що забезпечує унікальними адресами мережевого рівня значно більшу кількість пристроїв. Значною мірою, майбутнє інтернету речей не буде можливим без підтримки IPv6, отже, глобальне впровадження IPv6 матиме вирішальне значення для успішного розвитку інтернету речей.

Для бездротової передачі даних особливо важливу роль в побудові інтернету речей відіграють такі характеристики, як ефективність, відмовостійкість, адаптивність, можливість самоорганізації. Основний інтерес в цьому сенсі представляє стандарт IEEE 802.15.4, що управляє доступом для організації енергоефективних персональних мереж, і є основою для таких протоколів, як ZigBee та 6LoWPAN [3].

ZigBee – це комунікаційна технологія, заснована на протоколі IEEE 802.15.4 для реалізації низькошвидкісних бездротових приватних мереж. ZigBee має такі

характеристики, як низьке енергоспоживання, низька швидкість передачі даних, низька вартість і висока пропускна здатність. В даний момент ZigBee використовується в основному для обміну інформацією між різними речами електронного обладнання, які знаходяться на короткій відстані і мають не високу швидкість передачі даних. Це, в основному периферійні пристрої (миша, клавіатура) і побутова електроніка (TV, DVD), а також пристрої промислового управління (монітори, давачі і засоби автоматизації) [3].

Wi-Fi – це локальна бездротова технологія, яка використовує 2,4 ГГц надвисокої частоти або 5 ГГц супер-високочастотної радіохвилі. Ця технологія добре підходить для передавання великих обсягів даних по бездротовій мережі між пристроями, але це також вимагає багато енергії для роботи і має невеликий рівень пропускної здатності даних. При використанні цієї технології потрібно буде замінювати батареї у всіх пристроях на регулярній основі [3].

Bluetooth – це бездротова технологія, яка використовується для передачі даних в персональних мережах. Він передає дані по смузі частот від 2,4 до 2,485 ГГц і працює на коротших відстанях, ніж Wi-Fi. Ви можете синхронізувати пару пристроїв, таких як телефони, навушники, колонки, комп'ютери і багато іншого. З розвитком Bluetooth v4.0 з'явилася можливість реалізувати функцію низького енергоспоживання і збільшений радіус дії до декількох десятків метрів [3].

Серед провідних технологій важливу роль у розповсюдженні інтернету речей відіграють рішення PLC – технології побудови мереж передачі даних по лініях електропередач, оскільки у багатьох додатках присутній доступ до електромереж (наприклад, торгові автомати, банкомати, інтелектуальні лічильники, контролери освітлення спочатку підключені до мережі електропостачання). 6LoWPAN, який реалізує шар IPv6 як над IEEE 802.15.4, так і над PLC, будучи відкритим протоколом, стандартизованих IETF, відзначається як особливо важливий для розвитку інтернету речей [3].

1.2 Контролер ESP8266

ESP8266 – це мікропроцесорний контролер китайського виробника Espressif Systems, оснащений інтерфейсом Wi-Fi. Окрім наявності інтерфейсу Wi-Fi, мікроконтролер відрізняється тим, що не має вбудованої флеш-пам’яті в SoC і здатен виконувати програми з зовнішньої флеш-пам'яті з інтерфейсом SPI.

Мікроконтролер ESP8266 привернув увагу в 2014 році у зв'язку з виходом перших продуктів на його базі за неочікувано низькою ціною. Навесні 2016 року було розпочато виробництво ESP8285, що об'єднує ESP8266 та флеш пам'ять на 1 МБ. Восени 2015 року Espressif Systems запропонувала вдосконалену модель лінійки – мікросхему ESP32 [13].

До переваг застосування ESP8266 можна віднести те, що контролер отримав широке розповсюдження, має низьку вартість, існує достатньо велика кількість документації, навчальних матеріалів, засобів розробки, в тому числі, бібліотек для інших мікроконтролерних платформ.

ESP8266 пропонує повністю автономне мережеве рішення Wi-Fi, що дозволяє або виконувати програму, або використовувати функції мережі Wi-Fi спільно з іншим мікропроцесорним контролером. Коли ESP8266 виконує програму та є єдиним процесором на пристрої, програма може завантажуватися безпосередньо із зовнішньої флеш-пам’яті. Контролер має вбудований кеш для покращення продуктивності системи в таких програмах і мінімізації вимог до пам’яті. Крім того, завдяки можливості виконувати роль адаптера Wi-Fi, бездротовий доступ до Інтернету можна додати до будь-якого пристрою на основі мікроконтролера з простим підключенням через інтерфейс UART або інтерфейс моста CPU AHB [14].

Оскільки мікроконтролер не має на кристалі енергонезалежної пам'яті для користувача, виконання програми ведеться із зовнішнього постійного запам’ятовуючого пристрою (ПЗП) шляхом динамічного завантаження необхідних проміжків програми в кеш інструкцій. Завантаження виконується апаратно,

прозоро для розробника. Контролером підтримується до 16 МБ зовнішньої пам'яті програм. Можливий Standard, Dual або Quad SPI інтерфейс [13].

Виробник не надає документації на внутрішню периферію контролеру. Замість цього надано набір бібліотек, через API яких розробник отримує доступ до периферії. Оскільки ці бібліотеки інтенсивно використовують ОЗП контролера, то виробник у документах не вказує точну кількість ОЗП на кристалі, а надає лише приблизну оцінку тієї кількості пам'яті, що залишається після лінкування бібліотек і становить близько 50 кБ. Ентузіасти, що дослідили бібліотеки ESP8266, припускають, що він має 32 кБ кешу інструкцій та 80 кБ ОЗП даних [13].

Контролер ESP8266 виготовляється в корпусі QFN 5×5 з 32 виводами. Схема розташування виводів контролера ESP8266 наведена на рисунку 1.1.

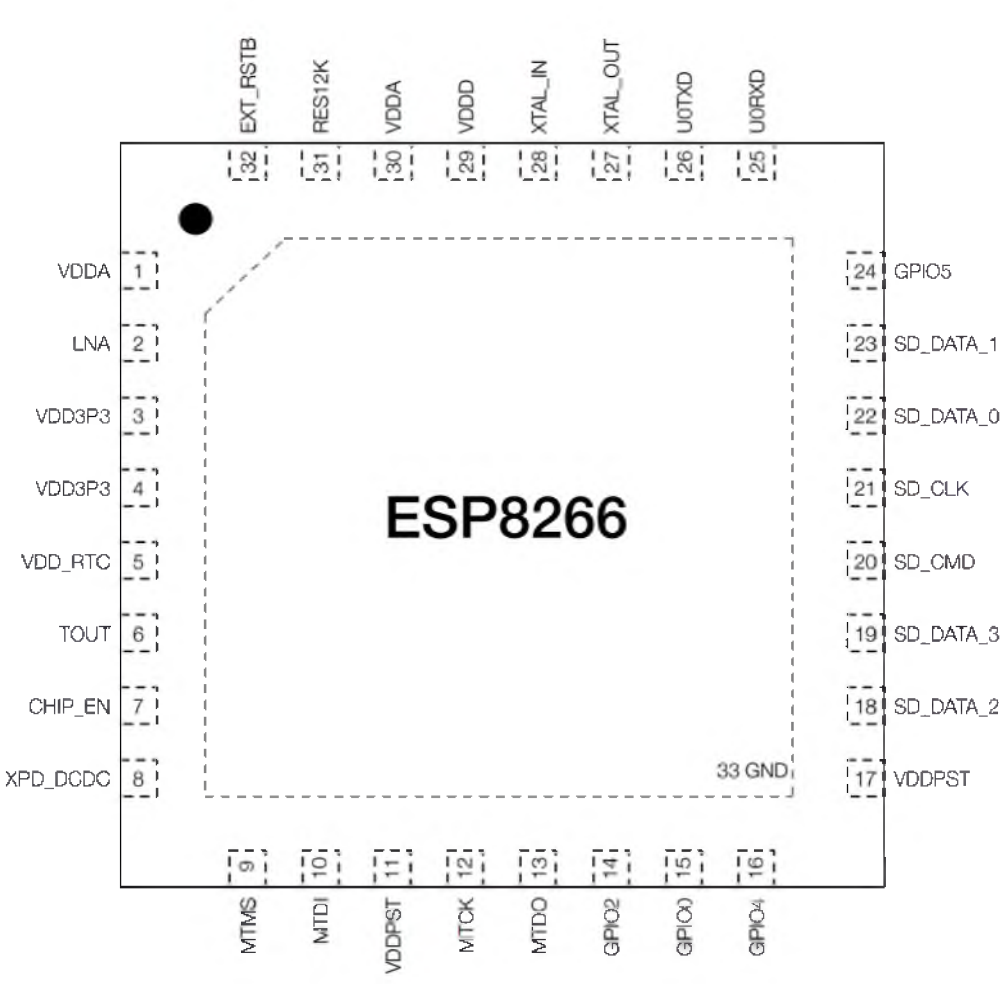

Рисунок 1.1 – схема розташування виводів контролера ESP8266.

Технічні характеристики [15] контролера ESP8266 наведені в таблиці 1.1.

Таблиця 1.1 – технічні характеристики контролера ESP8266

Робоча напруга	2,5…3,6 В
Середнє значення робочого струму	80 мА
Споживання струму в режимі передачі	120…170 мА
Споживання струму в режимі прийому	50…56 мА
Споживання струму в режимі Modem Sleep	15 мА
Споживання струму в режимі Light Sleep	0,9 мА
Споживання струму в режимі Deep Sleep	20 мкА
Діапазон робочих температур	-40…125 °C
Стандарти комунікації WLAN	IEEE 802.11 b/g/n Wi-Fi
Діапазон частот	2,4000…2,4835 ГГц
Підтримка інтерфейсів	UART, I^2C, I^2S, SPI, SDIO
Процесор	Tensilica L106 32-bit
Тактова частота процесора	80 МГц або 160 МГц
Порти вводу-виводу загального призначення	17 шт
Розрядність АЦП	10 біт
Загальний обсяг ОЗП	160 КБ
Пам’ять інструкцій	32 КБ
Пам’ять кешу	32 КБ
Пам’ять даних користувача	80 КБ
Пам’ять системних даних	16 КБ
Рекомендований типовий обсяг ПЗП	512 КБ…4 МБ
Максимальний обсяг ПЗП, що підтримується	16 МБ

Програмні засоби розробки (програмний комплект розробника, SDK) для ESP8266 складаються з [16]:

- Компілятора. Компілятор Xtensa LX106 входить до пакету компіляторів GNU Compiler Collection. Він має відкритий вихідний код. У різних SDK можуть міститися різні зборки цього компілятора, що трохи відрізняються підтримуваними опціями.
- Бібліотек до роботи з периферією контролера, стеків протоколів Wi-FI, TCP/IP.
- Засобів завантаження програми в пам'ять програм мікроконтролера.
- Опціональної IDE.

Espressif вільно розповсюджує свій комплект розробника. До цього комплекту входить компілятор GCC, бібліотеки Espressif та завантажувальна утиліта XTCOM. Бібліотеки постачаються у вигляді скомпільованих бібліотек, без

вихідних текстів. Espressif підтримує дві версії SDK: одна на основі RTOS, інша на основі зворотних дзвінків (callback) [16].

Крім офіційного SDK, існує ряд проектів альтернативних SDK, що використовують бібліотеки Espressif або пропонують власний еквівалент бібліотек Espressif, отриманий методами реверс-інжинірингу [16].

Щоб спростити використання мікроконтролера в типових проектах, можливе використання готових бінарних файлів, придатних до прямої заливки в ПЗУ модулів (прошивок). Готові прошивки можна розділити на кілька груп згідно з концепцією їх використання [16]:

- Прошивки для роботи під керуванням зовнішнього контролера. У цих прошивках реалізовано завдання параметрів через зовнішній контролер UART. До таких прошивок відноситься прошивка з керуванням AT-командами із SDK Espressif.
- Прошивки із вбудованими інтерпретаторами різноманітних мов високого рівня. Ці прошивки дозволяють завантажувати через UART та виконувати скрипти розробника пристрою.
- Прошивки для інтернету речей. Цей клас прошивок дозволяє з одного боку підключити до ESP8266 набір датчиків та виконавчих пристроїв, а з іншого боку надає необхідну функціональність мережі для роботи в інфраструктурі IoT.
- Прошивки для перехідника UART-WiFi.

Передбачена можливість оновлювати прошивку пристрою через Wi-Fi. Для цього флеш-пам’ять програм розділяється на кілька частин. Одна відводиться менеджеру прошивок, а дві інші під програму користувача. Коли треба оновити прошивку, новий образ завантажується у вільну частину флеш-пам'яті. Після перевірки цілісності завантаженого образу менеджер прошивок перемикає прапорець, після чого ділянка пам'яті зі старою прошивкою звільняється, а виконання коду йде з нової ділянки. Відповідно наступного разу оновлення завантажуватиметься у вільну ділянку пам'яті [16].

ESP8266 може працювати як у ролі точки доступу, так і кінцевої станції. При роботі в локальній мережі ESP8266 конфігурується в режим кінцевої станції. Для

цього пристрою необхідно встановити SSID Wi-Fi мережі та пароль доступу. Режим точки доступу зручний для початкового налаштування пристрою. В такому режимі пристрій видимий під час пошуку мереж у планшетах та комп'ютерах. Залишається підключитися до пристрою, відкрити HTML-сторінку конфігурування та встановити параметри мережі, після чого пристрій штатно підключиться до локальної мережі в режимі кінцевої станції [16].

Найпростіший спосіб взаємодії з модулем ESP8266 – передавати йому АТ-команду та отримувати відповідь. Набір АТ-команд – це спеціальний набір інструкцій, завдяки яким він може виконувати певні дії при їх отриманні і видавати в термінал результат виконання. Програма, звана процесор АТ-команд, вже встановлена в модулі ESP8266 і готова до прийому через послідовний порт. Ці команди починаються із символів «АТ» [17].

Коли модуль підключено до терміналу ПК або іншого пристрою, можна відправити через послідовний порт найпростішу команду – «АТ». У відповідь на неї модуль має надіслати відповідь «ОК».

На основі контролера ESP8266 створено велику кількість модулів для підключення до інших мікроконтролерних платформ або самостійного використання. Найбільш розповсюдженими серед них є ESP-01, ESP-07 та ESP-12.

ESP-01 є найпростішим серед існуючих модулів на базі ESP8266. Схема розташування виводів цього модуля наведена на рисунку 1.2.

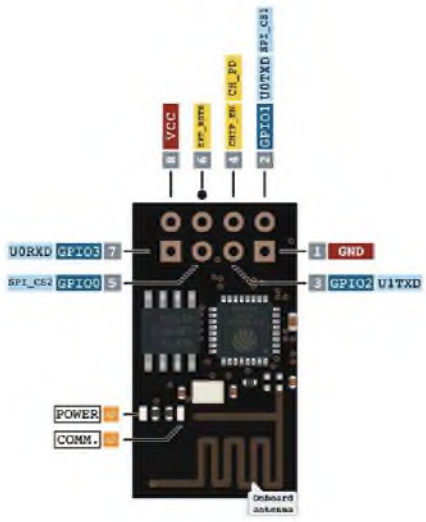

Рисунок 1.2 – Схема розташування виводів модулю ESP-01

Значно розширеними можливостями та підвищеною надійністю, порівняно з ESP-01, відрізняються модулі ESP-07 та ESP-12. Вони мають захисний екран і більшу кількість доступних для використання виводів. Кількість та розташування виводів цих модулів відрізняється незначно, але при цьому модуль ESP-07 має окрему керамічну антену, а в модулі ESP-12 антена виконана у вигляді доріжки на платі. Схема розташування виводів модулю ESP-12 наведена на рисунку 1.3.

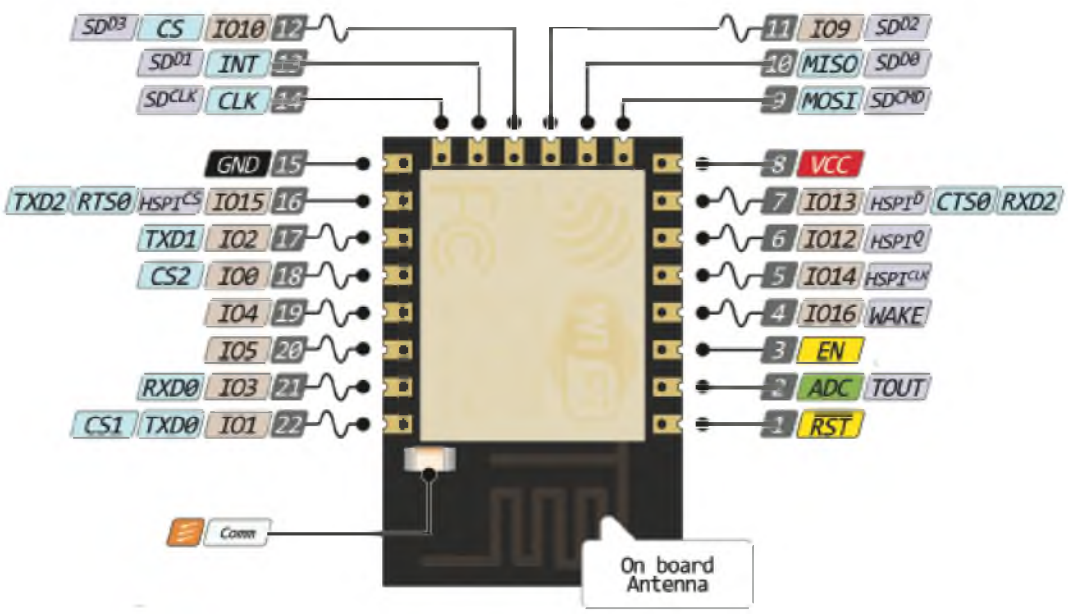

Рисунок 1.3 – Схема розташування виводів ESP-12

Такі модулі зручно використовувати в якості плат для відлагодження, макетування, створення прототипів пристроїв інтернету речей, але також можливе використання в остаточних версіях пристроїв.

У випадку розробки пристроїв розумного будинку, можливостей ESP8266 не завжди вистачає для виконання складних операцій, коли крім обміну даними з використанням мережі Wi-Fi пристрій має виконувати велику кількість інших дій. Цю проблему вирішує застосування додаткових мікроконтролерів. При цьому взаємодія контролерів здійснюється за допомогою AT-команд.

Контролер ESP8266 та модулі ESP часто застосовується спільно з іншими мікроконтролерами або платами на їх основі. Одним з найбільш розповсюджених є поєднання ESP8266 та мікроконтролерної платформи Arduino.

1.3 Мікроконтролерна платформа Arduino

Arduino – це апаратна обчислювальна платформа для аматорського конструювання, основними компонентами якої є плата мікроконтролера з елементами вводу/виводу та середовище розробки Processing/Wiring на мові програмування, що є спрощеною підмножиною C/C++. Arduino може використовуватися як для створення автономних інтерактивних об'єктів, так і підключатися до програмного забезпечення, яке виконується на комп'ютері. Інформація про плату (рисунок друкованої плати, специфікації елементів, програмне забезпечення) знаходяться у відкритому доступі [18].

Плата Arduino складається з мікроконтролера Atmel AVR, а також елементів обв’язки для програмування та інтеграції з іншими пристроями. На багатьох платах наявний лінійний стабілізатор напруги +5 В або +3,3 В. Тактування здійснюється на частоті 16 або 8 МГц кварцовим резонатором. У мікроконтролер записаний завантажувач (bootloader), тому зовнішній програматор не потрібен.

Плати Arduino дозволяють використовувати значну кількість виводів мікроконтролера як вхідні/вихідні контакти у зовнішніх схемах. Наприклад, у платі Decimila доступно 14 цифрових входів/виходів, 6 із яких можуть генерувати ШІМ сигнал, і 6 аналогових входів. Ці сигнали доступні на платі через контактні площадки або штирові роз'єми. Також існує багато різних зовнішніх плат розширення, які називаються «shields» («щити»), які приєднуються до плати Arduino через штирові роз'єми [18].

До основних переваг плат Arduino можна віднести [19]:

- Невисоку ціну – плати Arduino відносно недорогі порівняно з іншими платформами мікроконтролерів. Найдешевшу версію модуля Arduino можна зібрати вручну.
- Кросплатформенність – програмне забезпечення Arduino (IDE) працює в операційних системах Windows, Macintosh OSX і Linux.

– Просте, зрозуміле середовище програмування. Програмне забезпечення Arduino (IDE) просте у використанні для початківців, але досить гнучке, щоб досвідчені користувачі також могли скористатися ним.

– Програмне забезпечення з відкритим вихідним кодом і розширюване програмне забезпечення – ПЗ Arduino опубліковано як інструменти з відкритим кодом, доступні для розширення досвідченими програмістами. Мову можна розширити за допомогою бібліотек C++.

– Апаратне забезпечення з відкритим вихідним кодом і розширюване обладнання. Схеми плат Arduino опубліковані за ліцензією Creative Commons, тому досвідчені розробники схем можуть створити власну версію модуля, розширивши його та покращивши.

Існує багато версій плат Arduino, найбільш розповсюдженими з яких є Arduino Uno, Arduino Nano, Arduino Pro Mini та Arduino Mega. Ці плати побудовані на базі мікроконтролерів AVR та можуть бути застосовані у пристроях інтернету речей спільно з платами на базі ESP8266.

Розглянемо детальніше плату Arduino Uno. Загальний вигляд та схема розташування виводів плати наведені на рисунку 1.4.

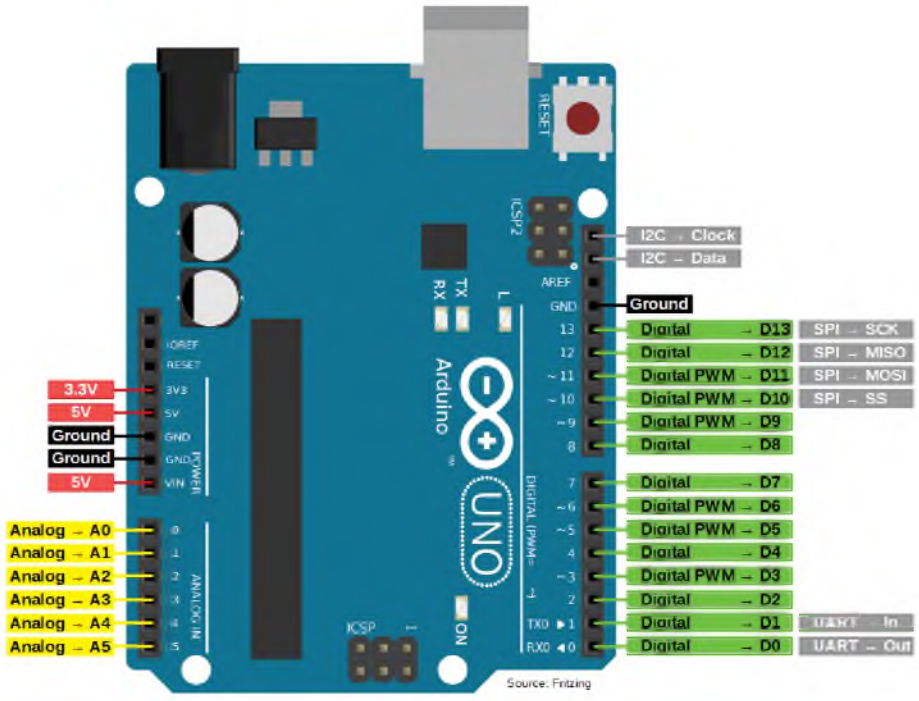

Рисунок 1.4 – Схема розташування виводів плати Arduino Uno

Технічні характеристики плат Arduino можуть відрізнятися залежно від версії. Але деякі характеристики, такі як архітектура, напруга живлення та сила струму у більшості випадків залишаються незмінними. У таблиці 1.2 наведено основні технічні характеристики плати Arduino Uno [20].

Таблиця 1.2 – Технічні характеристики плати Arduino Uno

Мікроконтролер	ATmega328p
Робоча напруга (логічний рівень)	5 В
Напруга живлення (рекомендована)	7…12 В
Напруга живлення (гранична)	6…20 В
Цифрові вводи/виводи	14 шт
Виводи з підтримкою ШІМ	6 шт
Аналогові вводи	6
Максимальний струм вводу/виводу	20 мА
Максимальний струм контакту 3,3 В	50 мА
Flash-пам'ять	32 КБ (0,5 КБ використовує bootloader)
SRAM	2 КБ
EEPROM	1 КБ
Тактова частота	16 МГц
Розрядність АЦП	10 біт
Підтримка інтерфейсів	UART, I^2C, SPI
Розміри плати	68,6 см × 53,4 см
Маса	25 г

Процес розробки програмного забезпечення для Arduino є порівняно простим. Для цієї мікроконтролерної платформи розроблено велику кількість бібліотек з відкритим вихідним кодом, найбільш важливі з яких можна встановити просто з середовища розробки – Arduino IDE. Мова програмування повністю сумісна з AVR C і дозволяє, за необхідності, розробляти більш оптимізоване програмне забезпечення з низьким використанням ресурсів та порівняно вищою швидкодією.

Як видно з наведених технічних характеристик, Arduino може взаємодіяти з ESP8266, використовуючи інтерфейси UART, I^2C та SPI. В найпростішому випадку для взаємодії використовується інтерфейс UART, що дозволяє контролерам обмінюватись даними через вбудований інтерфейс послідовного порту.

1.4 Застосування ESP8266 та Arduino у пристроях інтернету речей

Мікроконтролерна платформа Arduino в поєднанні з платами на основі ESP8266 часто застосовується для розробки пристроїв інтернету речей. Існують також плати Arduino, що вже містять у своєму складі одразу два мікроконтролери: AVR та ESP8266. Для таких плат Arduino створено хмарний сервіс інтернету речей Arduino Cloud. Для них також створені спеціалізовані бібліотеки та протоколи передачі даних, що дозволяють легко створювати пристрої інтернету речей без поглибленого вивчення двох окремих контролерів та без використання AT-команд.

Arduino IoT Cloud – це платформа, яка дозволяє будь-кому створювати проекти IoT із зручним інтерфейсом і єдиним рішенням для налаштування, написання коду, завантаження та візуалізації. Ця онлайн-платформа спрощує створення, розгортання та моніторинг проектів інтернету речей [21].

Для окремого застосування платформи Arduino та контролерів ESP8266 також створено велику кількість бібліотек та інших засобів розробки, що реалізують різноманітні протоколи передачі даних, які використовуються у пристроях інтернету речей, або навіть повноцінні операційні системи для пристроїв інтернету речей. Але такі рішення, у випадку поєднання плат Arduino та ESP, зазвичай потребують значно більших ресурсів, ніж застосування контролера ESP8266 без попередньої перепрошивки, оскільки для взаємодії двох контролерів достатньо використовувати AT-команди, що передаються через послідовний порт. Варто зазначити, що для роботи таких систем необхідне застосування перетворювачів логічних рівнів, оскільки робочі напруги ESP8266 та Arduino різні.

Розглянемо детальніше поєднання платформ Arduino та ESP, що взаємодіють за допомогою AT-команд, оскільки такі пристрої використовують мінімум сторонніх бібліотек та наочно реалізують взаємодію контролерів без необхідності попередньої перепрошивки контролера ESP8266.

Типове застосування ESP8266 як апаратної основи пристроїв інтернету речей найчастіше передбачає встановлення у будинках чи офісах. При цьому мережеве підключення здійснюється до домашньої/офісної локальної мережі з виходом в

інтернет через роутер. Користувач пристрою може контролювати його за допомогою планшета або комп'ютера через свою локальну мережу або віддалено через Інтернет [16].

Для простих пристроїв інтернету речей, що не потребують підключення до глобальної мережі (з метою дистанційного збору даних та/або управління користувачем, що перебуває за межами локальної мережі), достатньо використовувати контролер ESP8266 в режимі локального сервера. Але таке застосування контролера значно обмежує можливості взаємодії користувача з «розумним» пристроєм.

Після підключення до мережі Wi-Fi пристрій повинен отримати IP-параметри локальної мережі. Ці параметри можна встановити вручну разом з параметрами Wi-Fi або активізувати будь-які сервіси автоматичного конфігурування IP-параметрів. Після налаштування IP-параметрів звернення до сервера пристрою в локальній мережі зазвичай здійснюється за його IP-адресою, мережевим ім'ям (у разі, якщо імена підтримані якоюсь технологією) або сервісу (у разі, якщо підтримано автоматичний пошук сервісів) [16].

Часто доступ до пристрою потрібний з Інтернету. Наприклад, користувач з мобільного телефону віддалено перевіряє стан свого «розумного будинку», звертаючись безпосередньо до пристрою. У цьому випадку пристрій працює в режимі сервера, до якого звертається клієнт. Як правило, пристрій на основі ESP8266 знаходиться у локальній мережі офісу чи будинку. Вихід до Інтернету забезпечує роутер, підключений з одного боку до локальної мережі, а з іншого – до мережі провайдера Інтернету. Провайдер призначає роутеру свою статичну або динамічну IP-адресу і роутер здійснює трансляцію адрес локальної мережі в мережу провайдера. За замовчуванням правила цієї трансляції забезпечують вільну видимість інтернет-адрес з локальної мережі, але не дозволяють звернутися до локальних адрес з боку Інтернету. Існують наступні способів оминути це обмеження [16]:

- Конфігурація NAT. Більшість сучасних роутерів дозволяє встановити додаткові правила трансляції мережевих адрес між локальною та глобальною

мережами. Обидві технології дозволяють звернутися до сервера в локальній мережі з глобальної мережі, знаючи лише IP-адресу, видану роутеру провайдером. У разі статичної IP-адреси роутера це може бути задовільним рішенням для обмеженого кола користувачів системи. Однак такий підхід у більшості випадків незручний, оскільки необхідно вручну конфігурувати роутер і з'ясовувати IP-адресу роутера, яка може змінюватися регулярно.

- Динамічний DNS. Ці сервіси працюють як спеціальні сервери із фіксованими іменами в інтернеті. Розробник заводить на такому сервісі свій обліковий запис з унікальним ім'ям. Параметри цього облікового запису він прописує у пристрої. Пристрій у режимі клієнта періодично звертається до сервера сервісу, повідомляючи йому ім'я свого облікового запису та свою поточну IP-адресу. Кінцевий користувач в інтернеті звертається до цього ж сервісу та отримує від нього поточні IP-параметри пристрою. У такому випадку пристрій в мережі видно з доменним ім'ям третього рівня. Основна проблема DDNS-сервісів – це гарантії існування конкретного сервісу. Зазвичай, гарантується лише комерційний сервіс, коли його використання стягується плата.
- Зовнішні IoT-сервіси. Щоб полегшити проблему доступності пристрою в Інтернеті та зробити інсталяцію пристрою легкою для користувача, було розроблено низку рішень. Механізм цих рішень базується на існуванні в Інтернеті спеціального сервера, до якого може підключитися як IoT-пристрій, так і планшет/комп'ютер користувача. При цьому пристрій працює в режимі клієнта, ніяких спеціальних налаштувань роутера або особливих навичок користувача пристрою не потрібно. Обмін даними з пристроєм здійснюється за посередництва цього спеціального сервісу, параметри якого пристрій повинен закласти розробник.

З даного переліку можна зробити висновок, що взаємодія з пристроями інтернету речей за межами локальної мережі потребує застосування додаткових серверів. Отже, всі дані, що передаються між пристроєм та клієнтом користувача, будуть перенаправлені через зовнішній сервер. Тому важливим є забезпечення захисту даних, що передаються, та самих пристроїв, що отримують дані з Інтернету.

1.5 Аналіз вразливостей та загроз в системах IoT

Інтернет речей може викликати величезні зміни у повсякденному житті, надавши звичайним користувачам абсолютно новий рівень комфорту. Але якщо елементи такої системи не будуть належним чином захищені від несанкціонованого втручання, за допомогою надійного криптографічного алгоритму, замість користі вони принесуть шкоду, надавши кіберзлочинцям лазівку для підриву інформаційної безпеки. Оскільки речі із вбудованими комп'ютерами зберігають дуже багато інформації про свого власника, зокрема можуть знати його точне місцезнаходження, доступ до такої інформації може допомогти зловмисникам вчинити злочин. Недостатня опрацьованість на даний час стандартів для захисту таких автономних мереж дещо сповільнює впровадження інтернету речей у повсякденне життя [3].

У 2013 році були оприлюднені результати дослідження невідомим вченим загального стану безпеки в Інтернеті. Дослідження відбувались у 2012 році, дослідник перевіряв відкриті порти на всіх доступних IP-адресах. Через обсяг роботи, яку слід було виконати, дослідник створив комп'ютерного хробака, який шукав пристрої, доступ до яких не був захищений паролем, або захищений надзвичайно простим паролем (наприклад, «root» або «admin»). Створений ним ботнет, який отримав ім'я «Carna», зібрав понад 9 ТБ даних, виконав 52 мільйони запитів ICMP ping, 180 мільярдів службових записів, та 2,8 мільярди запитів TCP SYN на 660 мільйонів IP-адрес і опитав у сумі 71 мільярд портів. Його хробак спромігся поширитись на понад 400 тисяч пристроїв. В ході досліджень ним був помічений інший хробак, який отримав назву Aidra та був створений для пристроїв під управлінням ОС на основі Linux та процесорної архітектури MIPS. Основним призначенням хробака Aidra було створення ботнету для DDoS-атак. Всього було виявлено 30 тисяч заражених цим хробаком пристроїв. В 2013 році були оприлюднені у вільному доступі початкові коди хробака Aidra (LightAidra) [3].

У вересні 2016 року після публікації статті про угрупування, які продають послуги ботнетів для здійснення DDoS-атак, веб-сайт журналіста Брайана Кребса

сам став жертвою DDoS-атаки, трафік якої на піку досягав 665Гб/с, що робить її однією з найпотужніших відомих DDoS-атак. Оскільки хостер сайту відмовився надалі безоплатно надавати свої послуги, сайт довелось на деякий час закрити поки не був знайдений новий хостер. Атака була здійснена ботнетом з інфікованих «розумних» відео-камер (що є підмножиною інтернету речей). У жовтні того ж року зловмисники оприлюднили початкові тексти використаного шкідливого ПЗ (відоме під назвою Mirai), чим створили ризики неконтрольованого відтворення атак іншими зловмисниками [3].

Ботнет Mirai став можливим завдяки реалізації вразливості, яка полягала у використанні однакового, незмінного, встановленого виробником пароля для доступу до облікового запису адміністратора на «розумних» пристроях. Всього шкіливе ПЗ мало відомості про 61 різних комбінацій логін-пароль для отримання доступу до облікового запису методом перебору. Дослідження показали, що значна частина вразливих пристроїв була виготовлена з використанням складових виробництва фірм XiongMai Technologies та Dahua [22].

В п'ятницю, 21 жовтня 2016 року сталась потужна розподілена атака на відмову в обслуговуванні проти Dyn DNS, оператора DNS в США. Атака відбувалась у дві хвилі, що атакували сервери компанії, які знаходились в різних регіонах світу. Атака була підсилена спровокованим нею потоком повторних запитів (DNS retry) від мільйонів різних комп'ютерів з усього світу. Спровоковані запити через IP та UDP на порт 53 перевищували нормальний трафік у 40-50 крат (без врахування тих запитів, які не змогли дістатись серверів компанії внаслідок вжитих захисних заходів та перевантаження каналів зв'язку). Внаслідок атаки виникли проблеми із доступом до багатьох веб-сайтів, зокрема: Twitter, Etsy, Github, Soundcloud, Spotify, Heroku, та інші. Проведене компанією розслідування показало, що кістяк атаки спирався на близько 100 тисяч пристроїв типу «інтернет речей» керованих варіантом шкідливого ПЗ Mirai [22].

На початку листопада 2016 року створений на основі Mirai ботнет (Botnet 14) розпочав DDoS-атаку проти Ліберії. Країна отримала доступ до Інтернет завдяки єдиному оптоволоконному каналу ACE, прокладеному в 2011 році. Даний канал

надає зв'язок для всього західного узбережжя Африки та має пропускну здатність до 5,1 ТБ/с. Нападникам вдалось тимчасово розірвати доступ цілої країни до Інтернету [22].

В жовтні 2017 року інженери компанії Check Point оприлюднили доповідь про виявленого ними хробака, що атакує пристрої класу «інтернет речей» та утворює на їх основі ботнет. Новий хробак отримав назву IoTroop або Reaper. На відміну від Mirai, новий хробак покладається на щонайменше 9 відомих вразливостей у пристроях різних виробників. За оцінкою дослідників, новий хробак вразив пристрої у понад 1 млн організацій по всьому світу [22].

Крім порушення конфіденційності традиційних мереж зв'язку (повтори, підслуховування, спотворення інформації і т.д.), в системах інтернету речей виникають проблеми із захистом споживчої складової, що зумовлені [23]:

- відсутністю серйозного збитку;
- недостатньою кількістю стандартів не тільки захисту, але і взаємодії;
- недостатнім інтересом у виробників, як першої щаблі реалізації.

Велику загрозу несе керування пристроїв за допомогою міжмашинної взаємодії. Жодну написану людиною програму не можна вважати стовідсотково точною; для неї пишуться різні патчі для виправлення помилок. Така ж доля чекає датчики в інтернет-пристроях. Із посиленням ролі даних пристроїв в житті людей буде збільшуватися загроза безпеці всіх даних, навіть найнезначніших на перший погляд. Необхідно оцінювати будь-який витік інформації, так як резюмування її складових може представляти небезпеку для життя як фізичних, так і юридичних осіб (найбільших компаній) [23].

Одним з найбільш небезпечних напрямків атаки, на які варто звернути увагу, є DDoS-атака. Їх мета представляє з себе захоплення системних ресурсів і утруднення доступу до них інших користувачів [23].

У січні 2014 року в журналі Forbes кібержурналіст Джозеф Стейнберг опублікував список пов'язаних із Інтернетом приладів, які «шпигують» за нами буквально в наших будинках. До них належать телевізори, кухонна техніка, камери. Дуже ненадійна комп'ютерна система автомобілів, яка контролює гальма, двигун,

замки, капот, вентиляцію і приладову панель; ці частини системи найбільш уразливі для зловмисників під час спроби отримання доступу до бортової мережі. Також атака може бути проведена віддалено по Інтернету. Хакерами була продемонстрована можливість дистанційного керування електрокардіостимуляторами. Пізніше вони навчилися отримувати доступ до інсулінових помп й імплантованих кардіо-дефібриляторам [23].

У 2015 році компанія Hewlett Packard провела масштабне дослідження, в якому повідомляється, що 70% пристроїв інтернету речей мають вразливості своїх паролів, існують проблеми з шифруванням даних та з дозволами для доступу, і 50% додатків для мобільних пристроїв не обмінюються даними [23].

Згідно зі статтею [24], безпека інтернету речей залишається головним пріоритетом. Згідно з прогнозами GSMA, до 2025 року кількість підключень до IoT подвоїться і досягне майже 25 млрд в усьому світі, а в міру збільшення популярності IoT зростає ризик кібератак. Кібербезпека інтернету речей викликає занепокоєння у 95% респондентів опитування, проведеного аналітиками IoT Analytics, причому майже 40% «дуже стурбовані» можливими вразливостями інтернету речей. 88% вказали, що підтримують впровадження правил забезпечення безпеки IoT і прийняття галузевих стандартів для управління передовими методами кібербезпеки. Передбачається, що ринок безпеки IoT виросте до $ 36,6 млрд до 2025 року.

Багато пристроїв інтернету речей не оновлюються. Коли ранні комп'ютерні системи зіткнулися з тією ж проблемою, її (в деякій мірі) було виправлено за допомогою автоматичних оновлень програмного забезпечення. Але у випадку з пристроями IoT, довговічність не завжди є пріоритетом для виробників [24].

Виходячи з розглянутого вище, можна виділити наступні основні вразливості у системах інтернету речей:

- Ненадійна аутентифікація/авторизація;
- Відсутність або ненадійність шифрування;
- Вразливості каналів зв’язку (Wi-Fi, Bluetooth, ZigBee тощо);
- Вразливості програмного забезпечення.

1.6 Аналіз існуючих методів захисту IoT

Питання захисту систем інтернету речей розглядається у статтях [24–27]. Автори пропонують вирішення проблем кіберзахисту як на глобальних рівнях, таких як прийняття міжнародних стандартів кібербезпеки IoT, так і на рівні самих пристроїв інтернету речей, наприклад, необхідність шифрування. Розглянемо детальніше запропоновані існуючі засоби захисту систем інтернету речей.

Стаття [24] пропонує здійснювати моніторинг за допомогою ШІ, який може допомогти в прогнозуванні та запобігання атак в майбутньому. Експертам необхідно виявляти уразливості і усувати їх у міру їх появи. Сучасні хмарні сервіси вже використовують аналіз загроз для прогнозування проблем безпеки. Ми поступово наближаємося до деяких рішень безпеки для Інтернету речей, але для цього потрібен час.

У статті [25] розглянуто наступні рішення забезпечення кібербезпеки пристроїв інтернету речей:

- Комунікаційні технології. Шифрування сьогодні є прийнятною частиною будь-якого рішення для безпечного IoT. Але шифрування є складним і має наслідки від апаратного забезпечення до ключового управління. Проте час і витрати на розробку та експлуатацію безпечних комунікацій, достатніх для вирішення запланованих загроз, можна легко занизити.
- Послуги, мови та інструменти. Слабкі сторони програмного забезпечення у системі та коді є однією з трьох чудових дверей, що призводять до використання вразливостей в роботі IoT. Неадекватні аналізи загроз або вимоги, а також слабкі або невиконані процедури часто призводить до слабкості програмного забезпечення. Отримання та ретельне використання служб, пов'язаних із безпекою, мовами, стандартами дизайном та кодуванням, а також інструментами, які їх підтримують, можуть здаватися дорогими, доки вартість не призведе до серйозного порушення, якого можна було б запобігти.
- Сертифікація. Сертифікація безпеки може вимагатись для певного напрямку діяльності. Навіть якщо це не потрібно зараз, усвідомлюючи вимоги

щодо сертифікації та включення корисних елементів у практику розробки, вже зараз потрібно створювати безпечні продукти та потенційно підготувати їх до вимог сертифікації в майбутньому.

- Промислова кооперація. Боротьба з хакерами – це асиметрична війна. Співпраця щодо виявлення дефектів, відстеження та спільного використання розробників, навіть конкурентів, стала прийнятною практикою. Так NIST Cybersecurity Framework призвела до розробки основ для організації зусиль щодо впровадження та адаптації практик безпеки в організації.

Запропоновані рішення в цілому зводяться до необхідності застосування надійних алгоритмів шифрування, розробки якісного програмного забезпечення та своєчасного його оновлення, створення пристроїв відповідно до вимог стандартів кібербезпеки, своєчасне виявлення вразливостей та інформування про виправлені дефекти.

Стаття [26] пропонує наступні рекомендації щодо безпеки IoT:

- Усі дані, що збираються та інформація, яка зберігається, повинні бути враховані. Кожен фрагмент даних та інформації, що циркулює в системі IoT, повинен бути відповідно відображений. Це стосується не лише того, що збирають датчики та пристрої, розгорнуті в навколишньому середовищі, але також стосується будь-яких можливих облікових даних на серверах автоматизації або інших додатках IoT.

- Кожен пристрій, що підключається до мережі, повинен бути налаштований з урахуванням безпеки. Перед підключенням пристрою до мережі слід забезпечити безпечні налаштування. Це включає використання надійних комбінацій імені користувача та пароля, багатофакторну автентифікацію та шифрування.

- Стратегія безпеки організації повинна будуватися з урахуванням компромісу. Хоча уникати порушень і компромісів важливо, визнання того, що немає ідеального захисту від нових загроз, може допомогти у створенні протоколів пом'якшення наслідків, які можуть значно містити та зменшувати наслідки успішної атаки.

- Кожен пристрій повинен бути фізично захищений. Важливо також враховувати фізичну доступність пристроїв IoT. Якщо сам пристрій IoT не має фізичних засобів захисту від фальсифікацій, його слід тримати в обмеженому місці або закріпити відповідними замками або іншими інструментами. Наприклад, IP-камери можна підробляти безпосередньо, якщо до них дістанеться кіберзлочинець. Їм можна імплантувати шкідливе обладнання або програмне забезпечення, яке може спричинити збої в системі або поширити зловмисне програмне забезпечення.
- Додатки, структури та платформи IoT, які покладаються на технологію блокчейн, повинні регулюватися, постійно відстежуватися та оновлюватися, щоб запобігти будь-яким майбутнім використанням криптовалюти.
- Одним з найкращих способів захисту від викрадення даних є використання транспортного шифрування та стандартів, таких як TLS. Інший спосіб – використовувати різні мережі, які ізолюють різні пристрої.

У статті [12] розглядається захист систем IoBT. Оскільки це середовище є дуже неоднорідним з точки зору пристроїв, мережевих стандартів, платформ, зв'язку тощо, виникають проблеми довіри, безпеки та конфіденційності, коли суб'єкти поля бою обмінюються інформацією один з одним. Для вирішення цих проблем, запропоновано платформу для аудиту на базі блокчейну для IoBT і докладно описано її архітектурні компоненти, такі як рівень визначення поля бою, мережевий рівень, консенсус і сервісний рівень.

Також відомий стандарт кібербезпеки для пристроїв інтернету речей ETSI EN 303 645, що пропонує положення забезпечення безпеки систем інтернету речей та пов'язаних з ним послуг [27]:

- Не використовувати універсальні паролі за замовчуванням;
- Впровадити засіб для керування звітами про вразливості;
- Своєчасно оновлювати програмне забезпечення;
- Надійно зберігати конфіденційні параметри безпеки;
- Забезпечити безпечну комунікацію;
- Мінімізувати відкриті місця для атаки;

- Забезпечити цілісність програмного забезпечення;
- Забезпечити безпеку персональних даних;
- Зробити системи стійкими до збоїв;
- Дослідити телеметричні дані системи;
- Спростити видалення даних користувачів;
- Спростити установку та обслуговування пристроїв;
- Перевіряти введені дані.

Також, для захисту від загроз безпеці, пов'язаних із застарілим або уразливим програмним забезпеченням пристрою, пропонується вживати заходи, щоб забезпечити підключені пристрої ефективними стратегіями для оновлення програмного забезпечення. Завдяки стандартам SOTA, що означає «програмне забезпечення по повітрю», і FOTA (прошивка по повітрю), програмне забезпечення підключених пристроїв, налаштування та інше цифрове програмування можна оновлювати за допомогою бездротового підключення. Перевага, яку це приносить, полягає не лише в цілісності та економічній ефективності методу, але й у значному покращенні подолання проблем безпеки пристрою [8].

1.7 Постановка задачі

В роботі розглядається забезпечення кібербезпеки пристроїв на базі контролера ESP8266. Цей контролер в пристрої може бути один або в поєднанні з іншим контролером і взаємодіяти через один з доступних інтерфейсів.

Оскільки пристрої інтернету речей на базі контролера ESP8266 часто виконуються з використанням інших мікропроцесорних контролерів, а також у зв'язку з простотою розробки та сумісністю платформ, пропонується забезпечення кібербезпеки пристрою, побудованого на базі плати з контролером ESP8266 та мікроконтролерної платформи Arduino. В цьому випадку взаємодія контролерів здійснюється з використанням AT-команд через інтерфейс послідовного порту або за допомогою бібліотек. Запропоновані рішення будуть сумісними з більшістю

контролерів AVR в поєднанні з контролером ESP8266, а також для систем, в яких цей контролер взаємодіє з комп'ютером напряму через UART-перетворювач.

Забезпечення кібербезпеки пристроїв інтернету речей є актуальною задачею, оскільки такі системи часто пов'язані з конфіденційною інформацією, безпекою людей, а також можуть широко застосовуватись для виконання DDoS-атак, майнінгу криптовалют тощо.

Задача роботи – розглянути вразливості пристроїв на базі ESP8266 та запропонувати захист пристроїв інтернету речей, побудованих на базі мікроконтролерів ESP8266. Передбачається, що пристрій може не використовувати існуючі операційні системи або протоколи передачі даних для інтернету речей.

1.8 Висновки

У першому розділі було розглянуто поняття та концепцію інтернету речей, його коротку історію, класифікацію та застосування систем інтернету речей, наведено основні технології, які використовуються в таких системах.

Було проаналізовано контролер ESP8266, надано його короткий опис, проаналізовано розташування виводів та їх призначення, наведено основні технічні характеристики ESP8266, описано найбільш широко застосовувані модулі на базі цього контролера, такі як ESP-01, ESP-07 та ESP-12. Також наведено інформацію про мікроконтролерну платформу Arduino, що часто використовується в пристроях інтернету речей сумісно з контролером ESP8266. Проаналізовано застосування ESP8266 та Arduino в пристроях інтернету речей, можливості їх взаємодії.

Проаналізовано основні вразливості пристроїв інтернету речей та методи їх захисту. Основними вразливостями є: ненадійна аутентифікація/авторизація, відсутність або ненадійність шифрування, вразливості каналів зв'язку та вразливості програмного забезпечення. В якості їх вирішення пропонується підвищення надійності аутентифікації/авторизації, забезпечення шифрування для всіх даних, захист каналу зв'язку, своєчасне оновлення програмного забезпечення. На основі цих даних було поставлено задачу роботи.

2 СПЕЦІАЛЬНА ЧАСТИНА

2.1 Аналіз вразливостей та загроз у пристроях на базі ESP8266

Для того, щоб проаналізувати можливі вразливості та загрози для пристроїв інтернету речей, побудованих на базі контролера ESP8266, розглянемо більш детально відомі вразливості та загрози для пристроїв інтернету речей, враховуючи характеристики ESP8266.

З точки зору безпеки, слід враховувати середовище передачі сигналу: в бездротових мережах отримати доступ до переданої інформації набагато простіше, ніж у провідних мережах. Досить помістити антену в зоні дії [28].

Контролер ESP8266 використовує Wi-Fi для бездротової передачі даних. Отже, більша частина загроз та вразливостей, притаманних мережам, створених з застосуванням Wi-Fi, існує і для пристроїв інтернету речей, побудованих на базі цього мікроконтролера. Розглянемо детальніше відомі вразливості та загрози в мережах Wi-Fi.

Головна відмінність між провідними і бездротовими мережами пов'язано з наявністю неконтрольованої області між кінцевими точками бездротової мережі. Це дозволяє атакуючим, що знаходяться в безпосередній близькості від бездротовий структур, виробляти цілий ряд нападів, які неможливі в дротовому світі [29].

Згідно з матеріалами статті [30], до основних загроз в мережах Wi-Fi можна віднести: DoS-атаки, неправильне налаштування мережі, «випадкові асоціації» (коли ноутбук з Windows XP досить довірливо ставиться до всіх бездротових мереж або просто некоректно налаштований бездротовий клієнт автоматично асоціюється і підключає користувача до найближчої бездротової мережі), слабкі ключі шифрування, методи автентифікації з відомими вразливостями, інтерференції, сканування мережі.

При використанні бездротового доступу до локальної мережі загрози безпеки істотно зростають. Перелічимо нижче основні вразливості і загрози бездротових мереж [29]:

- Мовлення радіомаяка. Точка доступу включає з певною частотою широкомовний «радіомаяк», щоб оповіщати навколишні бездротові вузли про свою присутність. Ці широкомовні сигнали містять основну інформацію про точку бездротового доступу, включаючи, як правило, SSID, і запрошують зареєструватися бездротові вузли в даній області. Будь-яка робоча станція, що знаходиться в режимі очікування, може отримати SSID і додати себе в відповідну мережу. Мовлення радіомаяка є вродженою патологією бездротових мереж. Багато моделей дозволяють відключати частину цього мовлення, що містить SSID, щоб кілька ускладнити бездротове підслуховування, але SSID проте посилається при підключенні, тому все одно існує невелике вікно уразливості.

- Виявлення WLAN. Для виявлення бездротових мереж WLAN використовується, наприклад, утиліта NetStumber спільно з супутниковим навігатором глобальної системи позиціонування GPS. Дана утиліта ідентифікує SSID мережі WLAN, а також визначає, чи використовується в ній система шифрування WEP. Застосування зовнішньої антени на портативному комп'ютері уможливлює виявлення мереж WLAN під час обходу потрібного району або поїздки по місту. Надійним методом виявлення WLAN є обстеження офісної будівлі з переносним комп'ютером в руках.

- Підслуховування. Підслуховування ведуть для збору інформації про мережу, яку передбачається атакувати згодом. Перехоплювач може використовувати здобуті дані для того, щоб отримати доступ до мережевих ресурсів. Обладнання, що використовується для підслуховування в мережі, може бути не складніше того, яке застосовується для звичайного доступу до цієї мережі. Бездротові мережі за своєю природою дозволяють з'єднувати з фізичної мережею комп'ютери знаходилися безпосередньо в мережі. Це дозволяє підключиться до бездротової мережі, розташований в будівлі, людині, що сидить в машині на стоянці поруч з ним. Атаку за допомогою пасивного прослуховування практично неможливо виявити.

- Помилкові точки доступу в мережу. Досвідчений атакуючий може організувати неправдиву точку доступу з імітацією мережевих ресурсів. Абоненти,

нічого не підозрюючи, звертаються до цієї помилкової точки доступу і повідомляють їй свої важливі реквізити, наприклад аутентифікаційну інформацію. Цей тип атак іноді застосовують в поєднанні з прямим глушінням, щоб заглушити справжню точку доступу в мережу.

- Відмова в обслуговуванні. Повну паралізації мережі може викликати атака типу «відмова в обслуговуванні» (DoS). Мета будь-якої DoS-атаки полягає в створенні перешкоди при доступі користувача до мережевих ресурсів. Бездротові системи особливо сприйнятливі до таких атак. Фізичний рівень в бездротової мережі – абстрактний простір навколо точки доступу. Зловмисник може включити пристрій, що заповнює весь спектр на робочій частоті перешкодами і нелегальним трафіком, така задача не викликає особливих труднощів. Сам факт проведення DoS-атаки на фізичному рівні в бездротової мережі важко довести.

- Атаки типу «людина-в-середині». Атаки типу «людина-в-середині» виконуються на бездротових мережах набагато простіше, ніж на провідних, так як до провідної мережі потрібно реалізувати певний вид доступу. Зазвичай атаки «людина-в-середині» використовуються для порушення конфіденційності і цілісності сеансу зв'язку. Атаки «людина-в-середині» більш складні, ніж більшість інших атак: для їх проведення потрібна детальна інформація про мережу. Зловмисник зазвичай підміняє ідентифікацію одного з мережевих ресурсів. Зловмисник використовує можливість прослуховування і нелегального захоплення потоку даних з метою зміни його вмісту, необхідного для задоволення деяких своїх цілей, наприклад спуфінгу IP-адрес, зміни MAC-адреси для імітування іншого хоста і т.д.

- Анонімний доступ в Інтернет. Незахищені бездротові локальні обчислювальні мережі забезпечують хакерам найкращий анонімний доступ для атак через Інтернет. Хакери можуть використовувати незахищену мережу WLAN організації для виходу через неї в Інтернет, де вони будуть здійснювати протиправні дії, не залишаючи при цьому своїх слідів. Організація з незахищеною ЛОМ формально стає джерелом атакуючого трафіку, націленого на іншу

комп'ютерну систему, що пов'язано з потенційним ризиком правової відповідальності за заподіяну шкоду жертві атаки хакерів.

З даного переліку вразливостей можна зробити висновок, що для забезпечення кіберзахисту пристроїв інтернету речей, що використовують для передачі даних Wi-Fi, важливо забезпечити захист для всієї мережі, до якої вони підключені.

Деякі з наведених вище вразливостей не відносяться до пристроїв інтернету речей на базі контролера ESP8266, але можуть бути здійснені в мережі, до якої підключений такий пристрій. Наприклад, мовлення радіомаяка, виявлення WLAN та використання точки доступу Wi-Fi для анонімного доступу в Інтернет не становить прямої загрози для пристрою та даних, що він передає, але загрожує загальній безпеці локальної мережі.

Найбільшу небезпеку, пов’язану з передачею даних в мережі Wi-Fi, для пристроїв інтернету речей на базі контролера ESP8266 становить можливість перехоплення даних, що передаються, та DoS-атаки, що можуть тимчасово зробити пристрій недоступним.

У випадку використання мікроконтролера ESP8266 в якості єдиного контролера у пристрої інтернету речей, все програмне забезпечення має зберігатися у зовнішній флеш-пам’яті, оскільки ESP8266 не має вбудованої пам’яті програм. В цьому випадку деякі вразливості можуть створити більшу загрозу, оскільки всі операції виконує безпосередньо контролер ESP8266. В даному випадку значну небезпеку становлять DoS-атаки.

Розглянемо цю вразливість на прикладі двоколісної балансуючої платформи [31–33], побудованої на базі модуля ESP-07. Подібні колісні платформи знаходять достатньо широке застосування. Наприклад, схожим об’єктом є сігвей. На відміну від чотириколісних, такі платформи можуть мати більшу висоту, зберігаючи при цьому більшу маневреність та стійкість. Основний інтерес вони представляють для розробників систем автоматичного управління, оскільки це відмінний приклад нестійких систем, і їх дослідження дозволяє відпрацювати різні регулятори і методи управління, які потім можуть бути застосовані до інших об’єктів [33]. В

даному випадку автоматизована система керування (АСК) балансуючою платформою використовує контролер ESP8266 одразу для зв'язку з оператором та для розрахунку керуючого впливу, необхідного для балансування. У випадку DoS-атаки на такий пристрій, керування буде припинено і балансуюча платформа стане некерованою, а зростаюче навантаження на контролер значно вплине на якість автоматичного регулювання і може призвести до падіння платформи.

Оскільки пристрої інтернету речей можуть використовувати для своєї роботи хмарні сервіси або інші сервери, що зберігають інформацію в базах даних, важливою є перевірка даних, що передаються або приймаються, на відсутність потенційно небезпечного вмісту. Відомі випадки [34], коли певні послідовності символів виводили з ладу мобільні пристрої. Такі атаки називаються текстовими бомбами. Небезпечний текст може бути отриманий пристроєм інтернету речей від зловмисника, що має доступ до мобільного додатку, веб-клієнта або безпосередньо до бази даних, та передані на мобільний пристрій користувача.

Окремо варто розглянути можливість здійснення SQL-ін'єкцій та PHP-ін'єкцій. Такі вразливості можуть призвести до отримання несанкціонованого доступу до бази даних або сервера, з яким здійснює обмін даними пристрій інтернету речей на базі контролера ESP8266.

SQL ін'єкція – один з поширених способів злому сайтів та програм, що працюють з базами даних, заснований на впровадженні в запит довільного SQL-коду. Впровадження SQL, залежно від типу СКБД та умов впровадження, може дати можливість тій людині, що атакує, виконати довільний запит до бази даних (наприклад, прочитати вміст будь-яких таблиць, видалити, змінити або додати дані), отримати можливість читання та/або запису локальних файлів та виконання довільних команд на сервері. Атака типу впровадження SQL може бути можлива за некоректної обробки вхідних даних, що використовуються в SQL-запитах [35].

Найпростішим прикладом SQL-ін'єкції є відправлення даних, що починаються з одинарної лапки та містять SQL-код. Наприклад, якщо запит, що виконує логічну операцію порівняння або пошуку, міститиме код ' OR '1'='1, то він поверне значення TRUE в будь-якому випадку.

PHP-ін’єкція – один із способів злому веб-сайтів, що працюють на PHP, який полягає у виконанні стороннього коду на серверній стороні. Потенційно небезпечними функціями є [35]:

- eval(),
- preg_replace() (з модифікатором «e»),
- require_once(),
- include_once(),
- include(),
- require(),
- create_function().

PHP-ін'єкція стає можливою, якщо вхідні параметри приймаються і використовуються без перевірки [35].

В той час як стандарти SOTA і FOTA забезпечують своєчасне оновлення програмного забезпечення контролера, що дозволяє підвищити рівень безпеки пристрою інтернету речей, сама по собі функція «оновлення по повітрю» (OTA) є потенційно небезпечною, оскільки може бути використана кожним, хто матиме доступ до локальної мережі, в якій знаходиться пристрій. Таким чином з’являється можливість встановлення будь-якого програмного забезпечення до пристрою інтернету речей, в тому числі, шкідливого.

Відсутність аутентифікації/авторизації або недостатня надійність її алгоритмів часто призводять до отримання несанкціонованого доступу до пристроїв інтернету речей або ПЗ для взаємодії з ними. Це може дозволити контролювати пристрій або отримати доступ до конфіденційної інформації.

В контролерах ESP8266 також було знайдено вразливості, описані у [37]. Одна з них дозволяє вивести контролер з ладу, відправивши надто велике число кількості доступних методів аутентифікації при підключенні до точки доступу. При цьому відбувається переповнення буфера контролера, що призводить до збою. Інші вразливості пов’язані з протоколом EAP, що дозволяють викликати збій в роботі контролера або перехопити зашифровану сесію. Ці вразливості було виправлено в контролері ESP32, але для ESP8266 виправлень наразі не існує.

2.2 Оцінка рівня загроз

Для оцінки рівня кожної з розглянутих загроз, розглянемо порушення властивостей інформації, таких як цілісність, доступність та конфіденційність, для кожного окремого випадку. Відповідно до проведеного аналізу буде можливо виділити найбільш небезпечні вразливості. Дані наведено в таблиці 2.1, де К – конфіденційність, Ц – цілісність, Д – доступність.

Таблиця 2.1 – Оцінка рівня загроз

Вразливість	**Наслідок**	**Порушення**
1	**2**	**3**
Підслуховування	Можливість отримання доступу до конфіденційних даних, що передаються або приймаються пристроєм інтернету речей, маючи можливість підключення до мережі Wi-Fi	К
Помилкові точки доступу	Можливість перехоплення всього трафіку, його модифікація або знищення	К, Ц, Д
Відмова в обслуговуванні	DoS-атаки, спрямовані проти пристроїв інтернету речей, здатні частково або повністю вивести з ладу пристрій та припинити обмін даними	Ц, Д
Атаки типу «людина-в-середині»	Можливість повного перехоплення, модифікації або створення трафіку	К, Ц, Д
Текстові бомби	Помилки в роботі програмного забезпечення, що можуть ускладнити або унеможливити його використання	Д
SQL- та PHP-ін’єкції	Можливість отримання повного доступу до керування базою даних або сервером хмарної служби	К, Ц, Д
OTA	Встановлення модифікованої версії прошивки під виглядом оновлення програмного забезпечення	К, Ц, Д
Вразливості аутентифікації/авторизації	Несанкціонований доступ до пристрою або розширення прав	К, Ц, Д
Вразливості апаратного забезпечення	Повний доступ до даних, часткове або повне виведення пристрою з ладу	К, Ц, Д

З усіх вразливостей, наведених у таблиці 2.1, найбільшу загрозу становлять ті, що порушують конфіденційність, цілісність та доступність одночасно. При цьому порушення конфіденційності є більш небезпечним, порівняно з порушенням цілісності та доступності, оскільки дані, що передаються або приймаються пристроєм інтернету речей на базі ESP8266, можуть містити конфіденційну інформацію про користувача.

У випадку, коли пристрій може використовувати декілька користувачів, до вразливостей аутентифікації/авторизації також можна віднести недостатнє обмеження прав окремих користувачів, що дозволяє отримати більше можливостей, ніж їм необхідно.

Незважаючи на те, що SQL- та PHP-ін'єкції в першу чергу напрямлені на сервер, а не безпосередньо на пристрій інтернету речей, атаки можуть бути здійснені з його застосуванням, а також можуть бути напрямлені проти нього або інших пристроїв.

Отже, найбільшу загрозу для пристроїв інтернету речей на базі контролера ESP8266 становлять:

- Помилкові точки доступу
- Атаки типу «людина-в-середині»
- SQL- та PHP-ін'єкції
- «Оновлення по повітрю» (OTA)
- Вразливості аутентифікації/авторизації
- Вразливості апаратного забезпечення

Крім цього, варто враховувати, що підслуховування може призвести до порушення конфіденційності, атаки типу «відмова в обслуговуванні» здатні частково або повністю зробити неможливою роботу пристрою протягом деякого часу або повністю вивести його з ладу, а «текстові бомби» напрямлені безпосередньо на пристрій, з якого користувач здійснює керування пристроєм інтернету речей. Порівняно з вразливостями, що порушують одразу всі властивості інформації, вони є менш небезпечними, але все одно становлять значну загрозу для пристроїв мережі інтернету речей та її користувачів.

2.3 Методи підвищення безпеки

Оскільки контролер ESP8266 працює в мережах Wi-Fi, більшість загроз для пристроїв інтернету речей, розроблених на його основі, пов'язані з бездротовою передачею даних. Виходячи з цього, в першу чергу розглянемо детальніше методи підвищення безпеки в мережах Wi-Fi.

Для запобігання атакам, можливо обмежити видимість мережі Wi-Fi. Це забезпечить ускладнене виявлення мережі сторонніми пристроями і знизить ризик несанкціонованого проникнення. Для свого виявлення точка доступу розсилає кадри-маячки. Кожен такий кадр містить службову інформацію для підключення і, зокрема, присутній SSID (ідентифікатор бездротової мережі). У разі прихованого SSID це поле порожнє, тобто виявлення бездротової мережі є неможливим і не можна до неї підключитися, не знаючи значення SSID. Але всі станції в мережі, які підключені до точки доступу, знають SSID і під час підключення, коли розсилають Probe Request запити, вказують ідентифікатори мереж, наявні в їх профілях підключень. Прослуховуючи робочий трафік, з легкістю можна отримати значення SSID, необхідне для підключення до бажаної точки доступу [28].

Ще одним способом підвищення безпеки в мережах Wi-Fi є фільтрація MAC-адрес. В цьому випадку для підключення пристрою до точки доступу відбувається перевірка MAC-адреси пристрою. Цей спосіб не входить до стандарту IEEE 802.11. Фільтрацію можна здійснювати такими трьома способами [28]:

- Точка доступу дозволяє отримати доступ станціям з будь-якою MAC-адресою;
- Точка доступу дозволяє отримати доступ тільки станціям, чиї MAC-адреси є в довіреному списку;
- Точка доступу забороняє доступ станціям, чиї MAC-адреси є в «чорному списку».

Найнадійнішим з погляду безпеки є другий спосіб, хоча він не розрахований на підміну MAC-адреси, що легко здійснити зловмисникові [28].

Крім цього, у всіх мережах Wi-Fi важливою складовою безпеки є аутентифікація – видача певних прав доступу абоненту на основі наявного в нього ідентифікатора [28].

Стандарт IEEE 802.11 передбачає два методи аутентифікації [28]:

- Відкрита аутентифікація. Робоча станція робить запит аутентифікації, у якому присутня тільки MAC-адреса клієнта. Точка доступу відповідає або відмовою, або підтвердженням аутентифікації. Рішення ухвалюється на основі MAC-фільтрації, тобто це захист на основі обмеження доступу, що не є безпечним.
- Аутентифікація із загальним ключем. Необхідно налаштувати статичний ключ шифрування алгоритму WEP. Клієнт робить запит у точки доступу на аутентифікацію, на що отримує підтвердження, яке містить 128 байт випадкової інформації. Станція шифрує отримані дані алгоритмом WEP (виконується побітове додавання з модулем 2 даних повідомлення з послідовністю ключа) і надсилає зашифрований текст разом із запитом на асоціацію. Точка доступу розшифровує текст і порівнює з початковими даними. У разі збігу надсилає підтвердження асоціації, і клієнт вважається підключеним до мережі. Схема аутентифікації із загальним ключем вразлива до атак «людина-в-середині». Алгоритм шифрування WEP – це проста XOR-послідовність з корисною інформацією, отже, прослухавши трафік між станцією і точкою доступу, можна відновити частину ключа. IEEE почав розробки нового стандарту IEEE 802.11i, але через труднощі затвердження, організація WECA спільно з IEEE анонсували стандарт WPA. У WPA використовується TKIP (протокол перевірки цілісності ключа), який використовує вдосконалений спосіб керування ключами та покадрову зміну ключа.

Наступним методом захисту мереж Wi-Fi є шифрування. Контролер ESP8266 підтримує шифрування WEP, WPA та WPA2. Розглянемо їх детальніше.

WEP-шифрування є аналогом шифрування трафіку в провідних мережах. Використовується симетричний потоковий шифр RC4, який досить швидко функціонує. На сьогоднішній день WEP і RC4 не вважаються криптостійкими [28].

Існують два основних протоколи WEP [28]:

- 40-бітний WEP (довжина ключа 64 біта, 24 з яких – це вектор ініціалізації, який передається відкритим текстом);
- 104-бітний WEP (довжина ключа 128 біт, 24 з яких – це теж вектор ініціалізації); Вектор ініціалізації використовується алгоритмом RC4. Збільшення довжини ключа не призводить до збільшення надійності алгоритму.

WPA-шифрування замість уразливого RC4, використовує криптостійкий алгоритм шифрування AES. Можливе використання EAP (розширюваний протокол автентифікації).

Шифрування WPA має два режими:

- Pre-Shared Key (WPA-PSK) - кожен вузол вводить пароль для доступу до мережі;
- Enterprise – перевірка здійснюється серверами RADIUS.

WPA2-шифрування (IEEE 802.11i) прийнято у 2004 році, з 2006 року WPA2 повинно підтримувати все вироблене Wi-Fi обладнання. В даному протоколі застосовується RSN (мережа з підвищеною безпекою). Спочатку в WPA2 використовувався протокол CCMP (протокол блочного шифрування з кодом автентичності повідомлення і режимом зчеплення блоків і лічильника). Основою є алгоритм AES. Для сумісності зі старим обладнанням є підтримка TKIP і EAP з деякими його доповненнями. Як і в WPA є два режими роботи: Pre-Shared Key і Enterprise.

Найбільш поширена атака на мережу Wi-Fi, захищену протоколами WPA-PSK або WPA2-PSK, – це атака за словником. Протокол захисту WPA-PSK або WPA2-PSK використовує ключ попередньої сесії (PTK – Pairwise Transient Key), який, відповідно, складається з попереднього загального ключа (PSK – Pre-Shared Key) та п’яти інших параметрів, таких як SSID, Authenticator Nounce (ANounce), Supplicant Nounce (SNounce), Authenticator MAC-address (MAC-адреса точки доступу) та Supplicant MAC-address (MAC-адреса Wi-Fi-клієнта). Цей ключ надалі використовує шифрування між точкою доступу та клієнтом. Зловмисник, який прослуховує ефір, може перехопити всі п’ять параметрів, окрім PSK. PSK отримується завдяки використанню парольної фрази WPA-PSK, яку відправляє

користувач разом із SSID. Комбінація цих двох параметрів пересилається за стандартом формування ключа на основі пароля PBKDF2 (Password Based Key Derivation Function), який генерує 256-бітовий загальний ключ. У звичайній WPA-PSK/WPA2-PSK атаці за словником зловмисник може використовувати програмне забезпечення, яке виводить 256-бітний PSK для кожної парольної фрази й використовувати її з іншими параметрами, які було описано під час створення PTK. PTK буде використовуватися для перевірки контрольної суми (MIC – Message Integrity Check) в одному з пакетів handshake. Якщо вони збігатимуться, то парольна фраза в словнику буде правильною. Водночас використовуються вразливості протоколу аутентифікації користувачів – відкрито передачу ANounce, SNounce, MAC-адреси точки доступу і MAC-адреси Wi-Fi-клієнта. Якщо під час відтворення алгоритму аутентифікації відбудеться успішна авторизація користувача, значить обраний зі словника пароль є істинним й атака призвела до успішного злому мережі [38].

Для запобігання атакам цього типу необхідно встановлювати надійний пароль, що значно ускладнить або зробить неможливим процес підбору парольної фрази за словником і зробить атаку недоцільною. Надійність пароля є мірою ефективності пароля від вгадування або брутфорс атак. У своїй звичайній формі, він оцінює, скільки спроб зловмисникові, не маючи прямого доступу до пароля, потрібно, в середньому, щоб правильно вгадати його. Сила пароля – це функція, що враховує довжину, складність і непередбачуваність [39].

Згідно з матеріалами [40], опублікованими Microsoft, надійний пароль характеризується наступними властивостями:

- щонайменше 12 символів, але краще 14 символів;
- комбінація букв у верхньому регістрі, букв нижнього регістра, чисел і символів;
- не містить слів, які можна знайти у словнику або ім'я особи, символу, продукту чи організації.
- значно відрізняється від попередніх.
- легко запам'ятовується, але складно вгадується.

Іншим важливим методом захисту пристроїв в локальній мережі Wi-Fi є розподіл її на окремі мережі. Пристрої інтернету речей безпечніше не підключати до основної мережі Wi-Fi, до якої підключені інші пристрої, а використовувати окрему або гостьову мережу. Це дозволить знизити ризик перехоплення трафіку від інших пристроїв у разі злому пристрою на базі ESP8266.

Отже, для підвищення захисту мережі Wi-Fi, до якої підключено пристрій інтернету речей на базі ESP8266, рекомендовано використовувати окрему мережу, встановлювати налаштування, що включають обмеження видимості мережі, фільтрацію MAC-адрес за довіреним списком, ввімкнути шифрування WPA2 та встановити надійний пароль.

Для унеможливлення реалізації атак, що використовують «текстові бомби», SQL- або PHP-ін'єкції, необхідно перевіряти дані, отримані від користувачів та пристроїв на вміст комбінацій символів, притаманних атакам цього типу та виконувати їх екранування – додавання інших символів, що зроблять виконання запиту до бази даних або коду PHP неможливим. Також можливо застосувати кодування всього тексту, наприклад, в Base64.

Для того, щоб зробити неможливими атаки, спрямовані на пристрої інтернету речей на базі ESP8266, через функцію «оновлення по повітрю», необхідно не використовувати для оновлень прошивки звичайний метод OTA, що широко застосовується для пристроїв на базі цього контролера.

Захист інформації, що передає або приймає пристрій інтернету речей на базі контролера ESP8266, можна забезпечити шифруванням всього вхідного та вихідного трафіку надійним криптостійким алгоритмом.

Для захисту пристроїв інтернету речей на базі контролера ESP8266 від атак, пов'язаних з проблемами аутентифікації/авторизації, необхідно обов'язково реалізувати надійний алгоритм аутентифікації, що передбачає передавання та зберігання паролів тільки в зашифрованому вигляді, перевірку пароля на надійність при реєстрації нового користувача, а також, за необхідності, розподілити права користувачів. Такий захист значно ускладнить отримання несанкціонованого доступу до панелі керування пристроєм.

2.4 Забезпечення кібербезпеки пристроїв інтернету речей на базі ESP8266

Виходячи з проведеного аналізу існуючих вразливостей та загроз, а також методів підвищення безпеки, можливо сформулювати необхідні заходи з захисту пристроїв інтернету речей, побудованих на базі мікроконтролера ESP8266, та розглянути детальніше їх реалізацію.

Забезпечення кібербезпеки пристроїв інтернету речей на базі контролера ESP8266 пропонується почати з організації та налаштування мережі Wi-Fi, до якої вони будуть підключені для виходу в Інтернет або взаємодії між собою в межах локальної мережі.

В першу чергу рекомендується створити для таких пристроїв окрему (гостьову) мережу Wi-Fi, до якої будуть підключені тільки пристрої інтернету речей. Важливо, щоб в цій мережі не були присутні основні пристрої, з яких здійснюється доступ в Інтернет – комп'ютери, смартфони, планшети тощо, або щоб їх перебування в цій мережі було мінімальним і не передбачало передачі конфіденційних даних користувачів. Така ізоляція забезпечить захист інших пристроїв, що можуть передавати конфіденційні дані, від «підслуховування» у випадку, якщо один або декілька пристроїв інтернету речей будуть зламані або якщо до мережі буде отримано несанкціонований доступ з іншого пристрою.

Після виділення окремої мережі для пристроїв інтернету речей необхідно налаштувати фільтрацію MAC-адрес за довіреним списком. Для цього необхідно знати MAC-адреси пристроїв. У випадку, якщо адреса невідома, часто її можна побачити в панелі адміністратора точки доступу Wi-Fi. Застосування фільтрації MAC-адрес значно ускладнить несанкціоноване проникнення в мережу, оскільки для цього буде необхідно підмінити MAC-адресу пристрою, щоб вона співпадала з однією з довірених.

Наступним кроком в налаштуванні мережі має бути налаштування обмеження видимості мережі Wi-Fi, в якій знаходяться пристрої інтернету речей. Це дозволить ускладнити виявлення мережі і, як наслідок, несанкціоноване проникнення до неї недовірених пристроїв.

Найбільш надійним шифруванням в мережах Wi-Fi, що підтримує контролер ESP8266, є WPA2. Тому необхідно налаштувати точку доступу таким чином, щоб при підключенні до неї використовувалось саме таке шифрування. Це забезпечить більш надійний захист даних, що передаються в мережі та ускладнить отримання несанкціонованого доступу до неї.

Для ускладнення підбору пароля доступу до мережі Wi-Fi за словником, необхідно встановити надійний пароль, що містить понад 12 символів, складається з літер у верхньому та нижньому регістрі, цифр та символів, не містить слів, пов'язаних з користувачем, які можуть міститися у словнику, або які можна вгадати. Пароль необхідно регулярно змінювати.

Дотримання наведених вище рекомендацій дозволить знизити ризики, пов'язані з вразливостями каналу зв'язку. Для пристроїв інтернету речей, побудованих на базі контролера ESP8266, це значно підвищить рівень захисту, але не буде достатнім, оскільки не вирішує проблеми інших розглянутих вразливостей.

У пристроях інтернету речей, що мають виконувати певні алгоритми незалежно від обміну даними, рекомендується застосовувати, крім ESP8266, інший контролер, для того, щоб у випадку здійснення DoS або іншої атаки, спрямованої на пристрій, що може вивести з ладу ESP8266, інший контролер продовжив виконувати необхідні дії, навіть після повної втрати зв'язку.

Для захисту пристрою від вразливостей аутентифікації необхідно розробити надійний аутентифікаційний алгоритм, а також реалізувати перевірку надійності пароля під час встановлення. Перевірка надійності включає перевірку на довжину пароля (не менше 12 символів), перевірку наявності літер верхнього та нижнього регістру, щонайменше однієї цифри та одного символу. Ця перевірка відбувається за межами пристрою інтернету речей, наприклад, в додатку або веб-інтерфейсі, через який здійснюється керування пристроєм.

Запропоновано алгоритм аутентифікації, що дозволяє щоразу передавати нові аутентифікаційні дані, що значно ускладнить можливість визначення пароля у випадку успішного перехоплення пакету з аутентифікаційними даними. Для цього на пристрої інтернету речей (сервер) та на пристрої, з якого здійснюється керування

(клієнт), зберігається хеш пароля, вигаданого користувачем. Для здійснення аутентифікації, з клієнта до сервера відправляється запит аутентифікації, після чого сервер генерує випадковий текстовий рядок і відправляє його клієнту, а також зберігає його в ОЗП. Клієнт «підписує» цей рядок збереженим хешем пароля, генеруючи новий хеш, і відправляє отриманий рядок до сервера. Сервер виконує таку ж саму операцію зі збереженим в ОЗП рядком і перевіряє результат з хешем, отриманим від клієнта. Таким чином, щоразу для аутентифікації генеруватиметься новий хеш, з якого буде значно складніше визначити пароль.

Програмне забезпечення для ESP8266 можливо розробляти в середовищі Arduino IDE, використовуючи відповідну мову програмування. Розглянемо приклад «підпису» випадкового тексту хешем паролю. Для створення хешу MD5 застосовується бібліотека «MD5.h».

```
#include <MD5.h>
byte sessionStr[33];
byte password[] = (char)”dc647eb65e6711e155375218212b3964”;
void signStr(byte *str, char *signedStr) {
byte localStr[66];
for (int i = 0; i < 66; i++) {
        if (i < 32) localStr[i] = password[i];
        else localStr[i] = sessionStr[i - 32];
        localStr[i + 1] = '\0';
}
unsigned char* hash = MD5::make_hash((char *)localStr);
char *md5str = MD5::make_digest(hash, 66);
for (int i = 0; i < 66; i++) {
        signedStr[i] = md5str[i];
        signedStr[i + 1] = '\0';
}
free(hash);
free(md5str); }
```

Як видно з наведеного вище коду, для «підпису» випадкового символьного рядку здійснюється його додавання до хешу пароля, збереженого на обох пристроях, та подальше отримання нового хешу з утвореного рядка. Таким чином стає можливим зберігання, передача та перевірка пароля в зашифрованому вигляді.

Для забезпечення захисту даних, що передаються між пристроєм інтернету речей на базі ESP8266 та іншими пристроями, необхідно реалізувати алгоритм шифрування даних. Для цього можливе застосування алгоритму AES.

AES (Advanced Encryption Standard) – симетричний алгоритм блочного шифрування (розмір блока 128 біт, ключ 128/192/256 біт), фіналіст конкурсу AES і прийнятий як американський стандарт шифрування урядом США. Вибір припав на AES з розрахуванням на широке використання та активний аналіз алгоритму, як це було із його попередником, DES [41].

AES має фіксовану довжину у 128 біт, а розмір ключа може приймати значення 128, 192 або 256 біт. Через фіксований розмір блоку AES оперує із масивом 4 × 4 байт, що називається станом (версії алгоритму із більшим розміром блоку мають додаткові колонки) [41].

Для ключа 128 біт алгоритм має 10 раундів у яких послідовно виконуються операції [41]:

- subBytes()
- shiftRows()
- mixcolumns() (у 10-му раунді пропускається)
- xorRoundKey()

Розглянемо детальніше кожну з операцій, що виконується для шифрування.

Процедура SubBytes() обробляє кожен байт стану незалежно, проводячи нелінійну заміну байтів використовуючи таблицю замін (S-box). Така операція забезпечує нелінійність алгоритму шифрування. Побудова S-box складається з двох кроків. По-перше, проводиться отримання зворотного числа в полі Галуа $GF(2^8)$. По-друге, до кожного байту b з яких складається S-box застосовується така операція:

$$B'_i = b_i \oplus b_{(i+4)mod8} \oplus b_{(i+5)mod8} \oplus b_{(i+6)mod8} \oplus b_{(i+7)mod8} \oplus c_i, \quad (2.1)$$

де $0 \leq i < 8$, і де b_i є і-й біт b, а c_i – і-й біт константи $c = 63_{16} = 99_{10} = 01100011_2$. Таким чином, забезпечується захист від атак, заснованих на простих алгебраїчних властивостях. S-box можна відобразити таблицею простої підстановки [41].

ShiftRows() працює з рядками таблиці State. При цій трансформації рядка стану циклічно зсуваються на r байтів по горизонталі, залежно від номера рядка. Для нульового рядка r = 0, для першого рядка r = 1 і т. д. Таким чином кожна колонка вихідного стану після застосування процедури ShiftRows складається з байтів з кожної колонки початкового стану. Для алгоритму Rijndael патерн зсуву рядків для 128- і 192-бітних рядків однаковий. Однак для блоку розміром 256 біт відрізняється від попередніх тим, що 2-й, 3-й і 4-й рядки зміщуються на 1, 3, і 4 байти, відповідно. Фактично це проста перестановка байтів таблиці 4x4 State [41].

У процедурі MixColumns(), чотири байти кожної колонки State змішуються, використовуючи для цього зворотну лінійну трансформацію. MixColumns опрацьовує стан по колонках, трактуючи кожну з них як поліном четвертого степеня. Над цими поліномами виконується множення в $GF(2^8)$ по модулю $x^4 + 1$ на фіксований многочлен $c(x) = 3x^3 + x^2 + x + 2$. Разом з ShiftRows, MixColumns вносить дифузію в шифр. Під час цієї операції, кожен стовпчик множиться на матрицю, яка для 128-бітного ключа має вигляд:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \tag{2.2}$$

У процедурі AddRoundKey RoundKey кожного раунду об'єднується зі State. Для кожного раунду RoundKey виходить із CipherKey використовуючи процедуру KeyExpansion; кожен RoundKey такого ж розміру, що і State. Процедура виробляє побітовий XOR кожного байта State із кожним байтом RoundKey. Фактично це звичайний побайтовий XOR байт ключа з байтами таблиці State [41].

Для програмної реалізації даного алгоритму запропоновано використання бібліотеки «AESLib.h», доступної в Arduino IDE. Зразок коду шифрування та дешифрування:

```
#include <AESLib.h>
void setup() {
}
void loop() {
uint8_t key[] = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,
25,26,27,28,29,30,31}; // ключ
char data[] = "0123456789012345"; // дані
// шифрування
aes256_enc_single(key, data);
// розшифровування
aes256_dec_single(key, data);
}
```

Таким чином, всі дані, що передаються або приймаються пристроєм інтернету речей в мережі Wi-Fi, будуть зашифровані з використанням алгоритму AES, що забезпечить захист даних від прочитання у випадку підслуховування, підключення до підробленої точки доступу, атак типу «людина-в-середині» та від деяких вразливостей апаратного забезпечення.

2.5 Висновки

У другому розділі було проаналізовано основні вразливості пристроїв інтернету речей на базі контролера ESP8266, враховуючи його характеристики. Оскільки ESP8266 використовує для передачі даних мережі Wi-Fi, було досліджено основні вразливості цих мереж, оцінено рівень кожної з можливих загроз та описано методи захисту від них.

Надано рекомендації щодо захисту мереж Wi-Fi, до яких підключаються пристрої інтернету речей. Запропоновано використовувати для таких пристроїв гостьові мережі, обмежувати їх видимість, використовувати фільтрацію MAC-адрес. Обґрунтовано вибір шифрування в бездротовій мережі, обґрунтовано важливість вибору надійного пароля для точки доступу та його характеристики.

Обґрунтовано необхідність перевірки даних, що отримує та передає пристрій на наявність коду SQL або PHP, що може становити загрозу для серверу. Запропоновано використовувати екранування деяких символів або кодування всіх даних.

Запропоновано алгоритм аутентифікації, що значно ускладнює ймовірність перехоплення паролю або його підміни під час аутентифікації, оскільки аутентифікаційні дані, що передаються, щоразу змінюються. Для забезпечення конфіденційності було запропоновано використовувати шифрування AES, що дозволяє надійно захистити дані від розшифровування.

Таким чином було виконано задачу, поставлену у першому розділі.

3 ЕКОНОМІЧНА ЧАСТИНА

Метою економічного розділу є обгрунтування економічної доцільності забезпечення кіберзахисту пристроїв інтернету речей на базі контролера ESP8266. Захист систем інтернету речей потребує використання ресурсів, таких як часовий, людський та фінансовий. Забезпечення кіберзахисту пристроїв інтернету речей дозволить зменшити вірогідність збитків від кіберзагроз. Для аналізу економічної доцільності необхідно виконати розрахунок капітальних витрат, річних експлуатаційних витрат, річного економічного ефекту та показників економічної ефективності розробки.

3.1 Розрахунок трудомісткості забезпечення кіберзахисту

Трудомісткість забезпечення кібербезпеки пристроїв інтернету речей визначається тривалістю кожної робочої операції, починаючи зі складання технічного завдання і закінчуючи оформленням документації, за умови, що роботу виконує один спеціаліст з інформаційної безпеки.

$$t = t_{\text{тз}} + t_{\text{в}} + t_{\text{а}} + t_{\text{вз}} + t_{\text{озб}} + t_{\text{овр}} + t_{\text{д}}, \text{ год}, \quad (3.1)$$

де $t_{\text{тз}}$ – тривалість складання технічного завдання на запровадження кіберзахисту пристроїв інтернету речей на базі ESP8266;

$t_{\text{в}}$ – тривалість вивчення ТЗ, літературних джерел за темою;

$t_{\text{а}}$ – тривалість розробки блок-схеми алгоритму;

$t_{\text{пр}}$ – тривалість проектування апаратної частини та програмування;

$t_{\text{опр}}$ – тривалість опрацювання методики на ПК;

$t_{\text{д}}$ – тривалість підготовки технічної документації.

Умовна кількість операторів:

$$Q = q \cdot \text{c} \cdot (1 + p), \text{ чол}, \quad (3.2)$$

де q – очікувана кількість операторів;

c – коефіцієнт складності розроблення методики;

p – коефіцієнт корекції методів в процесі їх опрацювання.

Коефіцієнт складності розробки методики с визначає відносну складність виконання щодо типового завдання, складність якого дорівнює одиниці. Діапазон його зміни – 1,25…2,0. Коефіцієнт корекції рекомендацій р береться з діапазону 0,05…0,1, що відповідає внесенню 3…5 корекцій і переробці 5-10% готової програми.

Розрахуємо цей показник, припускаючи, що коефіцієнт складності с=1,3, а коефіцієнт рекомендацій р=0,05. Також припустимо, що для початкової розробки необхідно 3 людини: 1 інженер АСКТП, 1 розробник ПЗ, 1 спеціаліст з інформаційної безпеки.

$$Q = 3 \cdot 1{,}3 \cdot (1 + 0{,}5) = 6, \text{ чол},$$

Тривалість складання технічної документації на розробку методики оцінимо у 50 годин.

Тривалість вивчення технічного завдання, опрацювання довідкової літератури з урахуванням уточнення ТЗ і кваліфікацію виконання оцінюється за формулою:

$$t_{\text{в}} = \frac{Q \cdot B}{(75...85) \cdot k}, \text{ год}, \qquad (3.3)$$

у якій В – коефіцієнт збільшення тривалості етапу внаслідок недостатнього опису завдання. В=1,2…1,5;

k – коефіцієнт, що враховує кваліфікацію виконавця і визначається стажем роботи за фахом

Для розробки даної методики припустимо, що коефіцієнт В=1,2, а коефіцієнт, що враховує кваліфікацію виконавця, візьмемо рівним k=1, що відповідає 2-річному стажу роботи.

Тоді, за формулою 3.3, розрахуємо тривалість вивчення ТЗ:

$$t_{\text{в}} = \frac{6 \cdot 1{,}2}{80 \cdot 1} = \frac{7{,}2}{80} = 0{,}09, \text{ год},$$

Тривалість розробки блок-схеми алгоритму розраховується за наступною формулою:

$$t_{\text{а}} = \frac{Q}{(20...25) \cdot k}, \text{ год}, \qquad (3.4)$$

Оскільки в розробці методики приймають виконавці різних спеціалізацій з різною кваліфікацією, розрахуємо за формулою 3.4 тривалість розробки блок-схеми алгоритму для окремих груп.

Для спеціалістів АСКТП зі стажем роботи 2 роки:

$$t_{\text{а,а}} = \frac{2}{20 \cdot 1} = 0{,}1, \text{ год},$$

Розробники ПЗ зв стажем роботи 3 роки:

$$t_{\text{а,р}} = \frac{2}{20 \cdot 1{,}1} = 0{,}09, \text{ год},$$

Спеціалісти з інформаційної безпеки:

$$t_{\text{а,і}} = \frac{2}{20 \cdot 1} = 0{,}1, \text{ год},$$

Тоді загальна тривалість виконання блок-схеми становить:

$$t_{\text{а}} = t_{\text{а,а}} + t_{\text{а,р}} + t_{\text{а,і}}, \text{ год}, \quad (3.5)$$

$$t_{\text{а}} = 0{,}1 + 0{,}09 + 0{,}1 = 0{,}29, \text{ год},$$

Тривалість створення програми за готовою блок-схемою розраховується за формулою:

$$t_{\text{пр}} = \frac{Q}{(20 \ldots 25) \cdot k}, \text{ год}, \quad (3.6)$$

У створенні програми беруть участь спеціалісти АСКТП та розробники ПЗ.

$$t_{\text{пр,а}} = \frac{2}{20 \cdot 1} = 0{,}1, \text{ год},$$

$$t_{\text{пр,р}} = \frac{2}{20 \cdot 1{,}1} = 0{,}09, \text{ год},$$

Загальна тривалість створення програми за готовою блок-схемою складає:

$$t_{\text{пр}} = t_{\text{пр,а}} + t_{\text{пр,р}} = 0{,}1 + 0{,}09 = 0{,}19, \text{ год},$$

Тривалість опрацювання та тестування розраховується за наступною формулою:

$$t_{\text{опр}} = \frac{1{,}5 \cdot Q}{(4 \ldots 5) \cdot k}, \text{ год}, \quad (3.7)$$

$$t_{\text{опр,а}} = \frac{1{,}5 \cdot 2}{4 \cdot 1} = 0{,}75, \text{ год},$$

$$t_{\text{опр,р}} = \frac{1{,}5 \cdot 2}{4 \cdot 1{,}1} = 0{,}68, \text{ год},$$

$$t_{\text{опр,і}} = \frac{1{,}5 \cdot 2}{4 \cdot 1} = 0{,}75, \text{ год},$$

$$t_{\text{опр}} = 0{,}75 + 0{,}68 + 0{,}75 = 2{,}18, \text{ год},$$

Тривалість підготовки технічної документації розраховується також для кожної групи виконавців окремо:

$$t_{\text{д}} = \frac{Q}{(20...25)\cdot k} + \frac{Q}{(20...25)\cdot k} \cdot 0{,}75, \text{ год}, \quad (3.8)$$

$$t_{\text{д,а}} = \frac{2}{20\cdot 1} + \frac{2}{20\cdot 1} \cdot 0{,}75 = 0{,}175, \text{ год},$$

$$t_{\text{д,р}} = \frac{2}{20\cdot 1{,}1} + \frac{2}{20\cdot 1{,}1} \cdot 0{,}75 = 0{,}158, \text{ год},$$

$$t_{\text{д,і}} = \frac{2}{20\cdot 1} + \frac{2}{20\cdot 1} \cdot 0{,}75 = 0{,}175, \text{ год},$$

$$t_{\text{д}} = 0{,}175 + 0{,}158 + 0{,}175 = 0{,}508, \text{ год},$$

У таблиці 3.1 визначена тривалість виконання кожного етапу забезпечення кібербезпеки пристроїв інтернету речей на базі контролера ESP8266.

Таблиця 3.1

Змінна	Назва процесу	Тривалість, год
$t_{\text{тз}}$	Складання технічного завдання на запровадження кіберзахисту	50
$t_{\text{в}}$	Вивчення ТЗ, літературних джерел за темою	0,09
$t_{\text{а}}$	Розробка блок-схеми алгоритму роботи розробленої методики	0,29
$t_{\text{пр}}$	Програмування апаратного пристрою за блок-схемою	0,19
$t_{\text{опр}}$	Опрацювання роботи пристрою на ПК	2,18
$t_{\text{д}}$	Підготовка технічної документації та оцінка захищеності пристроїв інтернету речей на базі ESP8266	0,508

Отже, виходячи з отриманих даних, загальна трудомісткість виконання роботи з забезпечення кібербезпеки пристроїв інтернету речей на базі контролера ESP8266 становить:

$$t = 50 + 0{,}09 + 0{,}29 + 0{,}19 + 2{,}18 + 0{,}508 = 53{,}258, \text{ год},$$

3.2 Розрахунок витрат на впровадження методики

Витрати на створення програмного продукту $К_{ПЗ}$ складаються з витрат на заробітну плату розробника програмного забезпечення $З_{ЗП}$ і вартості витрат машинного часу, необхідного для опрацювання програми $З_{МЧ}$:

$$K_{ПЗ} = З_{ЗП} + З_{МЧ}, \text{ грн}, \quad (3.9)$$

Заробітна плата розробників враховує основну і додаткову заробітну плату, а також відрахування на соціальні потреби, визначається за формулою:

$$З_{ЗП} = t \cdot З_{пр}, \text{ грн}, \quad (3.10)$$

де t – загальна тривалість створення ПЗ, год;

$З_{пр}$ – середньогодинна заробітна плата з нарахунками грн/год.

Оскільки розробка методики виконується в межах України, де нормована тривалість робочого тижня не повинна перевищувати 40 год, вважатимемо, що тривалість робочого дня становить 8 год. Розрахуємо середньогодинну заробітну плату кожного виконавця за формулою:

$$З_{пр} = \frac{З_{СМ}}{20 \cdot t_{дн}}, \text{ грн/год}, \quad (3.11)$$

де $З_{СМ}$ – середньомісячна заробітна плата робітника, грн;

$t_{дн}$ – тривалість робочого дня.

Умовно приймемо заробітну плату інженера АСКТП – 10000 грн, розробника ПЗ – 20000 грн, спеціаліста з кібербезпеки – 15000 грн.

Розрахуємо середньогодинну заробітну плату кожного виконавця:

$$З_{пр,а} = \frac{10000}{20 \cdot 8} = 62{,}5, \text{ грн/год},$$

$$З_{пр,р} = \frac{20000}{20 \cdot 8} = 125, \text{ грн/год},$$

$$З_{пр,і} = \frac{15000}{20 \cdot 8} = 93{,}75, \text{ грн/год},$$

Розрахуємо заробітну плату кожного виконавця:

$$З_{ЗП,а} = 53{,}258 \cdot 62{,}5 = 3328{,}63, \text{ грн},$$

$$З_{ЗП,р} = 53{,}258 \cdot 125 = 6657{,}25, \text{ грн},$$

$$З_{ЗП,і} = 53{,}258 \cdot 93{,}75 = 4992{,}94, \text{ грн},$$

Отже, середня заробітна плата виконавця становить:

$$З_{ЗП} = \frac{З_{ЗП,а}+З_{ЗП,р}+З_{ЗП,і}}{3} = \frac{3328,63+6657,25+4992,94}{3} = 4992,94, \text{ грн.}$$

Вартість машинного часу для налагодження програми на ПК визначається за формулою:

$$С_{МЧ} = С_{МЧ} \cdot t_{опр} + t_д, \text{ грн,} \quad (3.12)$$

У якій $t_{опр}$ – трудомісткість налагодження програми;

$t_д$ – трудомісткість підготовки документації;

$С_{МЧ}$ – вартість 1 години машинного часу ПК:

$$С_{МЧ} = P \cdot C_e + \frac{Ф_{зал} \cdot Н_а}{F_p} + \frac{К_{ліц} \cdot Н_{ліц}}{F_p}, \text{ грн,} \quad (3.13)$$

У якій Р – встановлена потужність ПК, кВт. Р=1,5 кВт;

C_e – тариф на електричну енергію, грн/кВт·год. C_e=1,68 грн/кВт·год;

$Ф_{зал}$ – залишкова вартість ПК на поточний рік, грн;

$Н_а$ – річна норма амортизації на ПК, частки одиниці;

$Н_{ліц}$ – річна норма амортизації на ліцензійне ПЗ, частки одиниці;

$K_{ліц}$ – вартість ліцензійного ПЗ, грн;

$F_р$ – річний фонд робочого часу. $F_р$=1993 для 40-годинного робочого тижня.

Мінімально допустимий строк корисного використання $Т_а$ ПК складає 3 роки, тобто річна норма амортизації не має перевищувати:

$$Н_а = \frac{1}{Т_а} \cdot 100\%, \quad (3.14)$$

$$Н_а = \frac{1}{3} \cdot 100\% = 33,3\%,$$

Строк дії права користування ліцензійним ПЗ не може складати менш, ніж 1 рік, в такому випадку $Н_{ліц}$ не має перевищувати 100%.

Визначимо залишкову вартість одного ПК, як середню вартість наданих розробникам пристроїв (таблиця 3.1).

Таблиця 3.1 – Специфікація комп'ютерів

Кількість пристроїв	Назва пристрою	Специфікація	Вартість за пристрій
6	ПК	Процесор Intel Core i5-10400F (2.9–4.3 ГГц), SSD 512 ГБ, ОЗП 16 Гб, відеокарта AMD Radeon Vega 8.	19500 грн
Загальна вартість			117000 грн

Загальна вартість $К_{аз}$ становить 117000 грн. Середня балансова вартість ПК $Ф_{зал}$ становить 19500 грн.

Розрахуємо вартість ліцензійного ПЗ на один рік як загальну вартість таких ліцензій, які визначені у таблиці 3.3.

Таблиця 3.3 – Вартість ліцензійного ПЗ

Назва програмного продукту	Вартість ліцензії на 1 рік
Microsoft Windows 10 pro	204000 (6000) грн
Microsoft Office Professional 2020 Plus	4000 (500) грн
Загальна вартість	208000 грн

Таким чином, загальна вартість ліцензійного ПЗ $К_{ліц}$ дорівнює 208000 грн.

Визначимо вартість 1 години машинного часу пристроїв:

$$С_{МЧ} = 1{,}5 \cdot 1{,}68 + \frac{19500 \cdot 0{,}2}{1993} + \frac{208000 \cdot 1}{1993} = 109{,}502, \text{ грн},$$

Вартість машинного часу для розробки запропонованого методу:

$$З_{МЧ} = 109{,}502 + 2{,}18 + 0{,}508 = 112{,}19, \text{ грн},$$

Отже, вартість створення ПЗ $К_{ПЗ}$ є частиною одноразових капітальних витрат разом з витратами на придбання та налагодження апаратури:

$$К_{ПЗ} = 4992{,}94 + 112{,}19 = 5105{,}13, \text{ грн},$$

Капітальні витрати на проектування та впровадження проектного рішення розраховуються за формулою:

$$К = К_{пр} + К_{зпз} + К_{пз} + К_{аз} + К_{н}, \text{ грн}, \quad (3.15)$$

У якій $К_{пр}$ – вартість розробки проекту інформаційної безпеки та залучення для цього зовнішніх консультантів, 5000 грн;

$К_{зпз}$ – вартість закупівель ліцензійного основного й додаткового ПЗ, 208000 грн;

$К_{пз}$ – вартість створення основного й додаткового ПЗ, 5105,13;

$К_{аз}$ – вартість апаратного забезпечення, 117000 грн;

$К_{н}$ – витрати на встановлення обладнання та налагодження системи інформаційної безпеки, 10000 грн.

Таким чином, капітальні витрати складатимуть:

$$K = 5000 + 208000 + 5105{,}13 + 117000 + 10000 = 345105{,}13, \text{ грн}$$

3.3 Розрахунок експлуатаційних витрат

Експлуатаційні витрати – поточні витрати на експлуатацію та обслуговування об'єкта проектування за визначений період, що виражені у грошовій формі.

Річні експлуатаційні витрати на функціонування методики складають:

$$С = С_{в} + С_{к} + С_{ак}, \text{ грн}, \quad (3.16)$$

де $С_{в}$ – витрати на оновлення ПЗ;

$С_{к}$ – витрати на керування системою;

$С_{ак}$ – витрати, викликані активністю користувачів.

Витрати на керування складають:

$$С_{к} = С_{а} + С_{з} + С_{єв} + С_{ел} + С_{тос}, \text{ грн}, \quad (3.17)$$

де $С_{а}$ – річний фонд амортизаційних відрахувань;

$С_{з}$ – річний фонд заробітної плати інженерно-технічного персоналу;

$С_{ел}$ – вартість електроенергії, що споживається апаратурою;

$С_{тос}$ – витрати на технічне й організаційне адміністрування;

$С_{єв}$ – єдиний внесок соціального страхування.

Річний фонд амортизаційних відрахувань визначається у відсотках від суми капітальних інвестицій і дорівнює:

$$С_{а} = С_{а,аз} + С_{а,пз} = \frac{К_{аз}}{Т_{а}} + \frac{К_{зпз}}{Т_{а}}, \text{ грн}, \quad (3.18)$$

$$С_{а,аз} = \frac{117000}{5} = 23400, \text{ грн},$$

$$С_{а,пз} = \frac{208000}{5} = 41600, \text{ грн},$$

$$С_а = 23400 + 41600 = 65000, \text{ грн},$$

Річний фонд заробітної плати інженерно-технічного персоналу складає:

$$С_з = С_{осн} + С_{дод}, \text{ грн}, \quad (3.19)$$

де $С_{осн}$, $С_{дод}$ – основна і додаткова заробітна плата, грн на рік

Основна заробітна плата визначається, виходячи з місячного посадового окладу $З_{ЗП}$, а додаткова заробітна пата – в розмірі 8-10% від основної заробітної плати.

$$С_з = 4992,94 \cdot 12 + (4992,94 \cdot 0,1) \cdot 12 = 65906,81, \text{ грн},$$

Вартість електроенергії, що споживається, визначається за формулою:

$$С_{ел} = P \cdot F_p \cdot Ц_е, \text{ грн}, \quad (3.20)$$

У якій Р – встановлена потужність апаратури, 1,5 кВт·год;

F_p – річний фонд робочого часу системи, 1993;

$Ц_е$ – тариф на електроенергію – 1,68 грн/кВт·год.

$$С_{ел} = 1,5 \cdot 1993 \cdot 1,68 = 5022, \text{ грн},$$

Витрати на технічне й організаційне адміністрування та сервіс системи інформаційної безпеки визначається у відсотках від вартості капітальних витрат:

$$С_{тос} = К \cdot 0,02, \text{ грн}, \quad (3.21)$$

$$С_{тос} = 345105,13 \cdot 0,02 = 6902,1, \text{ грн},$$

Єдиний внесок $С_{єв}$ встановлюється в розмірі 22% на суму застрахованої заробітної плати.

$$С_{єв} = С_з \cdot 0,22, \text{ грн}, \quad (3.22)$$

$$С_{єв} = 65906,81 \cdot 0,22 = 14499,5, \text{ грн},$$

Розрахуємо витрати на керування системою за формулою 3.17:

$$С_к = 65000 + 65906,81 + 14499,5 + 5022 + 6902,1 = 157330,41, \text{ грн},$$

Отже, річні поточні витрати на функціонування системи складають:

$$С = 157330,41, \text{ грн}, \quad (3.16)$$

3.4 Оцінка можливого збитку

Визначимо загальний збиток внаслідок витоку конфіденційної інформації:

$$B = n \cdot A \cdot R, \text{ грн}, \quad (3.20)$$

де n – кількість записів в базі даних, що зазнали ураження

А – відшкодування за один втрачений запис; А = 1700 грн

Припустимо, що в результаті атаки на базу даних, яку використовував пристрій інтернету речей на базі контролера ESP8266, було втрачено 1000 конфіденційних записів.

$$B = 1000 \cdot 1700 \cdot 0{,}25 = 425000, \text{ грн},$$

Загальний ефект від впровадження захисту пристрою інтернету речей:

$$E = B - C, \text{ грн}, \quad (3.21)$$

$$E = 425000 - 157330{,}41 = 267669{,}41, \text{ грн},$$

3.5 Аналіз показників економічної ефективності

Економічна ефктивність забезпечення захисту пристроїв інтернету речей на базі контролера ESP8266 визначається за допомогою коефіцієнту повернення інвестицій та терміну окупності.

Коефіцієнт повернення інвестицій:

$$ROSI = \frac{E}{K}, \text{ частки одиниці}, \quad (3.22)$$

де Е – загальний ефект

К – капітальні інвестиції

$$ROSI = \frac{267669{,}41}{345105{,}13} = 0{,}77 \quad (3.22)$$

Термін окупності капітальних інвестицій показує, за скільки років капітальні інвестиції окупляться за рахунок загального ефекту:

$$T_o = \frac{K}{E} = \frac{1}{ROSI} = 1{,}3 \text{ , років} \quad (3.23)$$

3.6 Висновки

В економічному розділі було обґрунтовано економічну доцільність забезпечення кіберзахисту пристроїв інтернету речей на базі контролера ESP8266. Захист систем інтернету речей потребує використання ресурсів, таких як часовий, людський та фінансовий. Забезпечення кіберзахисту пристроїв інтернету речей дозволить зменшити вірогідність збитків від кіберзагроз. Для аналізу економічної доцільності було виконано розрахунок капітальних витрат, річних експлуатаційних витрат, річного економічного ефекту та показників економічної ефективності розробки.

Капітальні витрати становлять 345105,13 грн, загальна трудомісткість становить 53,258 год, термін окупності – 1,3 роки, а економічний ефект – 77%.

ВИСНОВКИ

В роботі було розглянуто поняття та концепцію інтернету речей, класифікацію та застосування систем інтернету речей, наведено основні технології, які використовуються в таких системах. Надано короткий опис контролера ESP8266, проаналізовано розташування та призначення виводів, наведено основні технічні характеристики. Також надано інформацію про мікроконтролерну платформу Arduino, що теж використовується в пристроях інтернету речей. Проаналізовано основні вразливості пристроїв інтернету речей та методи їх захисту. Основними вразливостями є: ненадійна аутентифікація/авторизація, відсутність або ненадійність шифрування, вразливості каналів зв'язку та вразливості програмного забезпечення, запропоновано їх вирішення.

Проведено аналіз основних вразливостей пристроїв інтернету речей на базі контролера ESP8266, враховуючи його характеристики. Оскільки ESP8266 використовує для передачі даних мережі Wi-Fi, було досліджено основні вразливості цих мереж, оцінено рівень кожної з можливих загроз та описано методи захисту від них. Надано рекомендації щодо захисту мереж Wi-Fi, до яких підключаються пристрої інтернету речей. Запропоновано використовувати для таких пристроїв гостьові мережі, обмежувати їх видимість, використовувати фільтрацію MAC-адрес. Обґрунтовано вибір шифрування в бездротовій мережі, обґрунтовано важливість вибору надійного пароля для точки доступу та його характеристики. Обґрунтовано необхідність перевірки даних, що отримує та передає пристрій на наявність коду SQL або PHP, що може становити загрозу для серверу. Запропоновано використовувати екранування деяких символів або кодування всіх даних.

Запропоновано алгоритм аутентифікації, що значно ускладнює ймовірність перехоплення паролю або його підміни під час аутентифікації, оскільки аутентифікаційні дані, що передаються, щоразу змінюються. Для забезпечення конфіденційності було запропоновано використовувати шифрування AES, що дозволяє надійно захистити дані від розшифровування.

ПЕРЕЛІК ПОСИЛАНЬ

1. Що таке кібербезпека? | Захисний комплекс Microsoft. URL: https://www.microsoft.com/uk-ua/security/business/security-101/what-is-cybersecurity (дата звернення 10.11.2022)

2. What is IoT? - Internet of Things Beginner's Guide – AWS. URL: https://aws.amazon.com/what-is/iot/ (дата звернення 10.11.2022)

3. Інтернет речей — Вікіпедія. URL: https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82_%D1%80%D0%B5%D1%87%D0%B5%D0%B9 (дата звернення 10.11.2022)

4. RFID — Википедия. URL: https://ru.wikipedia.org/wiki/RFID (дата звернення 11.11.2022)

5. Neil Gershenfeld, Raffi Krikorian, Danny Cohen. The Internet of Things. Scientific American. 2004. Т. 291, №4. С. 76-81

6. Интернет вещей — Википедия. URL: https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82_%D0%B2%D0%B5%D1%89%D0%B5%D0%B9 (дата звернення 12.11.2022)

7. Dave Evans. The Internet of Things. How the Next Evolution of the Internet Is Changing Everything. Cisco White Paper. Cisco Systems. 2011

8. What is Internet of Things (IoT)? Everything you need to know. URL: https://www.avsystem.com/blog/what-is-internet-of-things-explanation/ (дата заернення 15.11.2022)

9. Charith Perera, Chi Harold Liu, Srimal Jayawardena. The Emerging Internet of Things Marketplace From an Industrial Perspective: A Survey // IEEE Transactions on Emerging Topics in Computing. — 2015-12. — Т. 3, вип. 4. — С. 585–598. — ISSN 2168-6750. — doi:10.1109/tetc.2015.2390034.

10. Internet of Military Things — Wikipedia. URL: https://en.wikipedia.org/wiki/Internet_of_Military_Things (дата звернення 20.11.2022)

11. Internet of Battlefield Things (IoBT) REIGN. URL: https://iobt.illinois.edu/ (дата звернення 21.11.2022)

12. D. K. Tosh, S. Shetty, P. Foytik, L. Njilla and C. A. Kamhoua, "Blockchain-Empowered Secure Internet -of- Battlefield Things (IoBT) Architecture," MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), 2018, pp. 593-598, doi: 10.1109/MILCOM.2018.8599758.

13. ESP8266 — Вікіпедія. URL: https://uk.wikipedia.org/wiki/ESP8266 (дата звернення 22.11.2022)

14. ESP8266 Datasheet. URL: https://www.electroschematics.com/esp8266-datasheet/ (дата звернення 23.11.2022)

15. ESP8266EX Datasheet URL: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf (дата звернення 24.11.2022)

16. ESP8266 — Википедия. URL: https://ru.wikipedia.org/wiki/ESP8266 (дата звернення 25.11.2022)

17. Знакомимся с модулем ESP8266 подробнее. URL: https://hobbytech.com.ua/%D0%B7%D0%BD%D0%B0%D0%BA%D0%BE%D0%BC%D0%B8%D0%BC%D1%81%D1%8F-%D1%81-%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D0%B5%D0%BC-esp8266-%D0%BF%D0%BE%D0%B4%D1%80%D0%BE%D0%B1%D0%BD%D0%B5%D0%B5/ (дата звернення 26.11.2022)

18. Arduino — Вікіпедія. URL: https://uk.wikipedia.org/wiki/Arduino (дата звернення 27.11.2022)

19. What is Arduino? URL: https://www.arduino.cc/en/Guide/Introduction (дата звернення 28.11.2022)

20. Arduino Uno Rev3. URL: https://store.arduino.cc/products/arduino-uno-rev3 (дата звернення 30.11.2022)

21. Getting Started With the Arduino IoT Cloud. URL: https://docs.arduino.cc/arduino-cloud/getting-started/iot-cloud-getting-started (дата звернення 01.12.2022)

22. Mirai (ботнет) — Вікіпедія. URL: https://uk.wikipedia.org/wiki/Mirai_(%D0%B1%D0%BE%D1%82%D0%BD%D0%B5%D1%82) (дата звернення 02.12.2022)

23. Безпека інтернету речей — Вікіпедія. URL: https://uk.wikipedia.org/wiki/%D0%91%D0%B5%D0%B7%D0%BF%D0%B5%D0%BA%D0%B0_%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82%D1%83_%D1%80%D0%B5%D1%87%D0%B5%D0%B9 (дата звернення 03.12.2022)

24. Правило В. В., Хижняк С. П. Проблеми кібербезпеки інтернету речей. Інститут телекомунікаційних систем КПІ ім. Ігоря Сікорського, Україна. URL: http://conferenc.its.kpi.ua/2021/paper/viewFile/23204/12517 (дата звернення 05.12.2022)

25. Куник В. І., Базилевич В. М. Кібербезпека в умовах інтернету речей. Національний університет «Чернігівська політехніка». URL: http://ir.stu.cn.ua/bitstream/handle/123456789/22532/155-156%20%D0%9A%D1%83%D0%BD%D0%B8%D0%BA%20%D0%92.%20%D0%86..pdf?sequence=1&isAllowed=y (дата звернення 06.12.2022)

26. Опірський І.Р., Головчак Р.В. Мойсійчук І.Р. Балянда Т.С. Гаранюк С.П. Проблеми та загрози безпеці IoT пристроїв URL: https://csecurity.kubg.edu.ua/index.php/journal/article/view/231/205 (дата заернення 06.12.2022)

27. ETSI EN 303 645 V2.1.1 (2020-06). CYBER; Cyber Security for Consumer Internet of Things: Baseline Requirements. URL: https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.01_60/en_303645v020101p.pdf (дата звернення 07.12.2022)

28. Захист у мережах Wi-Fi — Вікіпедія. URL: https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%85%D0%B8%D1%81%D1%82_%D1%83_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0%D1%85_Wi-Fi (дата звернення 07.12.2022)

29. Панська А.В., Резніченко В.А. Загрози та вразливості бездротових мереж. Актуальні задачі та досягнення у галузі кібербезпеки. URL: http://dspace.kntu.kr.ua/jspui/bitstream/123456789/5022/1/AUConferenceCyberSecurity_November2016_p146.pdf (дата звернення 08.12.2022)

30. Татарчук Артем, Куперштейн Леонід, Лукічов Віталій. Класифікація загроз для Wi-Fi мереж. Комп'ютерні технології та Інтернет в інформаційному суспільстві. URL: https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/22563/86.%20%D0%9A%D0%BB%D0%B0%D1%81%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F%20%D0%B7%D0%B0%D0%B3%D1%80%D0%BE%D0%B7%20%D0%B4%D0%BB%D1%8F%20Wi-Fi%20%D0%BC%D0%B5%D1%80%D0%B5%D0%B6.%20%D0%A2%D0%B0%D1%82%D0%B0%D1%80%D1%87%D1%83%D0%BA%2C%20%D0%9A%D1%83%D0%BF%D0%B5%D1%80%D1%88%D1%82%D0%B5%D0%B9%D0%BD%2C%20%D0%9B%D1%83%D0%BA%D1%96%D1%87%D0%BE%D0%B2.pdf?sequence=1&isAllowed=y (дата звернення 09.12.2022)

31. Попко О.С., Лосіхін Д.А. Розробка автоматизованої мікропроцесорної системи регулювання і управління станом рухомої платформи // Четвертий том збірника тез доповідей IX Міжнародної науково-технічної конференції студентів, аспірантів та молодих вчених «Хімія та сучасні технології». Дніпро. – 2019. – Т. IV. – С. 35–36.

32. Шульгін О.Л., Лосіхін Д.А. Моделювання системи управління балансуючою платформою // Wissenschaftliche Ergebnisse und Errungenschaften: 2020: der Sammlung wissenschaftlicher Arbeiten «ΛΌГОΣ» zu den Materialien der internationalen wissenschaftlich-praktischen Konferenz – Мюнхен. – 2020. – Т. 1. – С. 132–134. – ISBN 978-3-471-37237-1.

33. Шульгін О.Л., Ляшенко О.А. Розробка моделі балансуючої платформи в середовищі візуального моделювання // Збірник тез доповідей VI Міжнародної науково-технічної конференції «Комп'ютерне моделювання та оптимізація складних систем» – Дніпро. – 2020. – С. 67–68.

34. iPhone ламають одним повідомленням: виявлено баг, який дозволяє вивести з ладу гаджет. URL: https://bykvu.com/ua/bukvy/iphone-lomajut-odnim-soobshheniem-obnaruzhen-bag-kotoryj-pozvoljaet-vyvesti-iz-stroja-gadzhet/ (дата звернення 10.12.2022)

35. SQL-ін'єкція — Вікіпедія. URL: https://uk.wikipedia.org/wiki/SQL-%D1%96%D0%BD%27%D1%94%D0%BA%D1%86%D1%96%D1%8F (дата звернення 11.12.2022)

36. PHP-ін'єкція — Вікіпедія. URL: https://uk.wikipedia.org/wiki/PHP-%D1%96%D0%BD%27%D1%94%D0%BA%D1%86%D1%96%D1%8F (дата звернення 12.12.2022)

37. ESP32/ESP8266 Wi-Fi Attacks. URL: https://github.com/Matheus-Garbelini/esp32_esp8266_attacks (дата звернення 13.12.2022)

38. Гарист А.В. Аналіз захищеності Wi-Fi мереж. Інформатика, обчислювальна техніка та автоматизація. URL: https://tech.vernadskyjournals.in.ua/journals/2021/2_2021/part_1/18.pdf (дата звернення 14.12.2022)

39. Надійність пароля — Вікіпедія. URL: https://uk.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B4%D1%96%D0%B9%D0%BD%D1%96%D1%81%D1%82%D1%8C_%D0%BF%D0%B0%D1%80%D0%BE%D0%BB%D1%8F (дата звернення 15.12.2022)

40. Створення та використання надійних паролів - Підтримка від Microsoft. URL: https://support.microsoft.com/uk-ua/windows/створення-та-використання-надійних-паролів-c5cebb49-8c53-4f5e-2bc4-fe357ca048eb (дата звернення 16.12.2022)

41. Advanced Encryption Standard — Вікіпедія. URL: https://uk.wikipedia.org/wiki/Advanced_Encryption_Standard (дата звернення 17.12.2022)

ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи

№	Формат	Найменування	Кількість листів	Примітки
1	А4	Реферат	2	
2	А4	Список умовних скорочень	2	
3	А4	Зміст	2	
4	А4	Вступ	1	
5	А4	Стан питання. Постановка задачі	29	
6	А4	Спеціальна частина	20	
7	А4	Економічна частина	11	
8	А4	Висновки	1	
9	А4	Перелік посилань	5	
10	А4	Додаток А	1	
11	А4	Додаток Б	1	
12	А4	Додаток В	1	
13	А4	Додаток Г	1	

ДОДАТОК Б. Перелік документів на оптичному носії

1. Пояснювальна записка Шульгін О.Л.pdf
2. Презентація Шульгін О.Л.pdf

ДОДАТОК В. Відгук керівника економічного розділу

Відгук керівника економічного розділу:

Керівник розділу _____ _____
(підпис) (прізвище, ініціали)

ДОДАТОК Г. Відгук керівника дипломної роботи

В І Д Г У К

на кваліфікаційну роботу Шульгіна Олексія Леонідовича
студента групи 125м-21з-1
на тему: «Забезпечення кібербезпеки пристроїв інтернету речей на базі контролера ESP8266»

Пояснювальна записка складається зі вступу, трьох розділів і висновків, викладених на 81 сторінці.

Метою кваліфікаційної роботи є аналіз існуючих вразливостей та загроз для пристроїв інтернету речей на базі контролера ESP8266 і розробка методів підвищення їх кібербезпеки.

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності магістра спеціальності 125 «Кібербезпека». Для досягнення поставленої мети в кваліфікаційній роботі вирішуються наступні задачі: аналіз вразливостей пристроїв інтернету речей на базі контролера ESP8266, методів їх захисту, пропозиція рішень щодо підвищення захисту пристроїв інтернету речей на базі контролера ESP8266.

Практичне значення результатів кваліфікаційної роботи полягає у підвищенні кіберзахисту автоматизованих систем керування, в тому числі пристроїв інтернету речей, побудованих на базі контролера ESP8266 або з його застосуванням.

Оформлення пояснювальної записки до кваліфікаційної роботи виконано з незначними відхиленнями від стандартів.

Недоліком роботи є недостатньо деталізований нечіткість сформульованих висновків в підрозділах та розділах роботи.

Загалом за час дипломування Шульгін Олексій Леонідович проявив себе фахівцем, здатним самостійно вирішувати поставлені задачі та заслуговує присвоєння кваліфікації магістр за спеціальністю 125 Кібербезпека, освітньо-професійна програма «Кібербезпека».

Рівень запозичень у кваліфікаційній роботі не перевищує вимог "Положення про систему виявлення та запобігання плагіату".

Кваліфікаційна робота заслуговує оцінки «добре»/78б.

Керівник кваліфікаційної роботи
д.ф-м.н., проф. Т.С. Кагадій
Керівник спеціального розділу
асистент Ю. А. Мілінчук