

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“___” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“ Програмне забезпечення системи кібербезпеки комп’ютерної
мережі розумного будинку ”

Виконав здобувач вищої освіти
IV курсу, групи КБ-22-2
ОПП «Кібербезпека»
спеціальності 125 «Кібербезпека»
_____ Д.А. Жабокрицький
«___» _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Р.М. Минайленко
«___» _____ 2026 р.

Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітньо-кваліфікаційний рівень бакалавр
Галузь знань 12 Інформаційні технології
Спеціальність 125 “Кібербезпека”
Освітньо-професійна (освітньо-наукова програма) “Кібербезпека”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
О.А.Смірнов
“ ” 20 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Жабокрицького Дмитра Анатолійовича

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи кібербезпеки
комп'ютерної мережі розумного будинку

керівник роботи Минайленко Роман Миколайович, канд. техн. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 27.02.2026 року № 129-02

2. Строк подання студентом роботи 1 .06.2026 р.

3. Мета та завдання випускної кваліфікаційної роботи Створення програмного
забезпечення системи кібербезпеки комп'ютерної мережі розумного будинку

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно
розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію

6. Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

6. Дата видачі завдання « » 20 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної бакалаврської роботи	Строк виконання етапів кваліфікаційної бакалаврської роботи	Примітка
1.	Аналіз існуючих систем керування	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.05.2026 р.	
8.	Попередній захист роботи	21.05.2026 р.	

Студент _____
(підпис) (прізвище та ініціали)

Керівник роботи _____
(підпис) (прізвище та ініціали)

Анотація

Жабокрицький Д. А. “Програмне забезпечення системи кібербезпеки комп’ютерної мережі розумного будинку”

125 Кібербезпека. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

У випускній кваліфікаційній бакалаврській роботі розглянуто питання створення програмного забезпечення системи кібербезпеки комп’ютерної мережі розумного будинку. Створена система контролює стан пристроїв через Інтернет, коли оновлення інформації на веб-сервері зчитується за певним алгоритмом, а потім система відповідає на відповідні інструкції з високим ступенем захисту. Користувач може безпосередньо входити в систему та взаємодіяти із вбудованим пристроєм у режимі реального часу. Система гнучка для розміщення широкого спектру вимірювальних приладів з відповідними інтерфейсами. Вона має різноманітні переваги, такі як енергоефективність, інтелект, низька вартість, портативність та висока продуктивність.

Проведені дослідження дозволяють зрозуміти, які програмні та апаратні засоби необхідні для побудови системи кібербезпеки комп’ютерної мережі розумного будинку. А також те, як це зробити з мінімальними матеріальними вкладеннями.

Розроблене програмне забезпечення дозволяє створити систему кібербезпеки комп’ютерної мережі розумного будинку.

Ключові слова: розумний будинок, система кібербезпеки, комп’ютерна мережа.

Abstract

Zhabokrytskyi D. A. “Software for the cybersecurity system of a smart home computer network”

125 Cybersecurity. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

The final qualifying bachelor's thesis considers the issue of creating software for the cybersecurity system of a smart home computer network. The created system monitors the state of devices via the Internet, when information updates on the web server are read according to a certain algorithm, and then the system responds to the corresponding instructions with a high degree of protection. The user can directly log in to the system and interact with the built-in device in real time. The system is flexible for accommodating a wide range of measuring devices with appropriate interfaces. It has various advantages, such as energy efficiency, intelligence, low cost, portability and high performance.

The conducted research allows us to understand what software and hardware are needed to build a cybersecurity system of a smart home computer network. And also how to do it with minimal material investments.

The developed software allows you to create a cybersecurity system of a smart home network.

Keywords: smart home, cybersecurity system, computer network.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	
ВСТУП.....	
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	
1.1 Призначення системи.....	
1.2 Область застосування.....	
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	
2.2 Обґрунтування вибору засобів для побудови системи моніторингу та мови програмування	
2.3 Розгорнута постановка завдання.....	
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	
3.1 Опис функціонування системи	
3.2 Розробка структурної схеми.....	
3.3 Розробка функціональної схеми	
3.4 Розробка діаграми процесів	
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	
4.2 Захист розробленого програмного забезпечення.....	
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	
6 ОСНОВНІ ВИСНОВКИ.....	
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	

					ВКБР-125.26.0028.00.00.ПЗ			
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив	Жабокрицький Д				Програмне забезпечення системи кібербезпеки комп'ютерної мережі розумного будинку	Літ.	Аркуш	Аркушів
Перевірів	Минайленко Р					Д	1	
Н. Контр.	Коваленко А.С					ЦНТУ КБ-22-2		
Затв.	Смірнов О.А.							

ПЕРЕЛІК СКОРОЧЕНЬ, СИМВОЛІВ І СПЕЦІАЛЬНИХ ТЕРМІНІВ

РБ – розумний будинок;
RFID – радіочастотна ідентифікація;
TCP – Transmission Control rotocol;
ПЗ – програмне забезпечення;
DVR – Digital Video Recorder;
RCE – Remote Command xecution;
SSL – Secure Sockets Layer;
OTA – On the air;
SSH – Secure Shell
JTAG – Joint Test Action Group;
SWD – Serial Wire Debug;
SPI – Serial Peripheral Interface;
UART – Universal Asynchronous Receiver-Transmitter;
CLI – Command Line Interface;
UDP – User Datagram Protocol;
DoS – Denial of Service;
IRC – Internet Relay Chat;
P2P – peer-to-peer;
Telnet – Teletype network;
C&C – Command and Control;
FTP – File transfer protocol;
DHCP – Dynamic Host Configuration Protocol;
EOM – Електронно-бчислювальна машина;
AV – antivirus.

					ВКБР-125.26.0028.00.00.ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		2

ВСТУП

Актуальність теми. Актуальність теми дослідження пов'язана з тим, що компанії, які займаються розробкою і впровадженням розумних будинків на даний час не виробили чіткої методики щодо їх створення і експлуатації. А подібним системам мають бути властиві такі показники як надійність, економічність, та захищеність. Тому тут відкривається безліч різноманітних можливостей і під час їх розробки, і під час експлуатації.

На теперішній час інтернет-технології все більше і більше входять в людське сьогодення і в промисловості, і в побуті. Потяг людей до комфортного побуту і ще й з мінімальними витратами часу на свої рутинні побутові проблеми призвело до виникнення низки різноманітних розумних систем. Власне одним із представників таких систем є розумний будинок (РБ).

Розумний будинок в сучасному розумінні стає об'єктом або зоною підвищеного комфорту в побуті людини, так як одна частина процесів в таких системах виконується автоматично, а іншою можна керувати віддалено з допомогою мобільних додатків встановлених в телефоні користувача. Такі системи швидко поширюються і стають більш доступними, оскільки дозволяють значно економити особистий час користувачів, зменшувати витрати на вирішення побутових проблем та підвищити рівень комфорту життя в таких приміщеннях.

Всі пристрої РБ об'єднані в комп'ютерну мережу. Однак системного підходу до вирішення проблеми кібербезпеки комп'ютерних мереж не вироблено, оскільки рішення про ввімкнення/вимкнення пристроїв ініціюється часто власником вручну. Крім того, існуючі системи розумного будинку часто вимагають додаткового мережевого обладнання, такого як шлюзи, маршрутизатори, а це призводить до додаткових вразливостей і знижує надійність систем кібербезпеки комп'ютерної мережі розумного будинку

Мета й завдання дослідження.. Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного

забезпечення системи кібербезпеки комп'ютерної мережі розумного будинку. Така система має бути простою в користуванні, забезпечувати захист комп'ютерної мережі розумного будинку, мати простий і зрозумілий інтерфейс.

Для досягнення поставленої мети потрібно:

- провести огляд існуючих систем кібербезпеки комп'ютерної мережі розумного будинку;
- розробити структурну схему системи;
- розробити функціональну схему системи;
- розробити діаграму взаємодії процесів системи кібербезпеки;
- розробити алгоритм роботи програмного забезпечення;
- створити програмну реалізацію системи кібербезпеки комп'ютерної мережі розумного будинку.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно реалізувати кібербезпеки комп'ютерної мережі розумного будинку.

Розробка контролює стан комп'ютерної мережі розумного будинку через Інтернет, коли оновлення інформації на веб-сервері зчитується за певним алгоритмом, що надходить в мікроконтролер, а потім система відповідає на відповідні інструкції з високим ступенем захисту. Користувач може безпосередньо входити в систему та взаємодіяти із системою у режимі реального часу. Система є енергоефективною, інтелектуальною, портативною та має високу продуктивність.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Розумний будинок є об'єктом, де всі пристрої функціонально пов'язані між собою комп'ютерною мережею, що дозволяє проводити моніторинг та контролювати їх за допомогою комп'ютера (або мікроконтролера) та забезпечувати віддалений доступ до них через Інтернет. Всі системи та пристрої розумного будинку узгоджують свої функції використовуючи інформаційні технології, шляхом порівняння заданих програм і зовнішніх показників.

Об'єкт розумний будинок керується при допомозі пульта дистанційного керування (ПДК), і господар має змогу створити власні умови комфорту, натисканням всього кількох кнопок. Система здатна сама аналізувати зовнішні та внутрішні параметри приміщень і, за результатами аналізу, виконувати команди налаштування, задані господарем розумного будинку.

На сьогоднішній день у компанії-розробники володіють різноманітним інструментарієм для кібербезпеки та моніторингу комп'ютерних мереж РБ. Однак системного підходу до вирішення проблеми кібербезпеки комп'ютерних мереж не вироблено, оскільки рішення про ввімкнення/вимкнення пристроїв ініціюється часто власником вручну. Крім того, існуючі системи розумного будинку часто вимагають додаткового мережевого обладнання, такого як шлюзи, маршрутизатори, а це призводить до додаткових вразливостей і знижує надійність систем кібербезпеки комп'ютерної мережі розумного будинку.

Важливою складовою системи є сертифіковані протоколи обміну інформацією між апаратними елементами. Така система повинна бути бездротовою, захищеною від радіоперешкод і небажаного зовнішнього втручання, надійною та енергоефективною, мати невисоку вартість і

можливість керування з комп'ютера (мікроконтролера) або мобільного телефону.

У моїй кваліфікаційній бакалаврській роботі запропоновано програмне забезпечення яке дозволяє створити систему кібербезпеки комп'ютерної мережі розумного будинку, яке певною мірою враховує вищезначені недоліки існуючих систем.

1.2 Область застосування

Створене програмне забезпечення системи кібербезпеки комп'ютерної мережі розумного будинку дозволяє контролювати стан пристроїв РБ через Інтернет, коли оновлення інформації на веб-сервері зчитується за певним алгоритмом, що надходить в мікроконтролер (Raspberry Pi), а потім система відповідає на відповідні інструкції з високим ступенем захисту. Користувач може входити в систему та комунікувати із пристроями мережі у режимі реального часу при допомозі мобільного пристрою (телефона). Система є інтелектуальною, енергоефективною, портативною, забезпечує високу продуктивність та має достатній рівень захищеності.

					ВКБР-125.26.0028.00.00.ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		6

2. ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Важливою частиною IoT-систем є кібербезпека комп'ютерних мереж. Як кіберфізична структура інтелектуальна мережа, заснована на Інтернеті Речей, стикається з декількома проблемами:

- підміна особистості (імперсонації) - вид атаки, націленої на несанкціоновану зв'язок від імені легітимного пристрою з використанням його ідентифікаційних даних. Зловмисник може підмінити ідентифікаційні дані одного з розумних лічильників, щоб маніпулювати даними про витрату енергії;
- несанкціоноване вилучення інформації. Оскільки об'єкти (пристрої) в інтелектуальній мережі, заснованій на Інтернеті Речей, зв'язуються, часто використовуючи загальну інфраструктуру зв'язку, зловмисник може легко отримати доступ до переданих даних;
- втручання в дані - зловмисник може змінити дані обміну, такі, як динамічні значення цін, відправлені в попередні пікові періоди (роблячи їх набагато менше). Наслідками цього можуть бути збільшення витрат домоволодінь замість їх зменшення, таким чином перенагружаючи енергосистему;
- проблеми авторизації і контролю доступу. З тих пір, як деякі пристрої можуть контролюватися і налаштовуватися віддалено (наприклад, розумні лічильники або територіально розміщені датчики і приводи в розподільних підстанціях), зловмисник або навіть незадоволений працівник може спробувати несанкціоновано отримати права доступу, щоб використовувати їх, пошкоджуючи фізичні об'єкти (такі, як трансформатори) або приводячи до відключень електроенергії;
- проблеми приватності. Розумні лічильники і розумні прилади в житлових будинках можуть повідомляти не тільки витрата енергії. Їх згенерувала

дрібномодульних інформація може порушити приватність кінцевого споживача за рахунок розголошення інформації про його звички (час підйому, сну або обіду і т.д.), про те, чи знаходиться він вдома або ж у відпустці тощо;

– компрометації і шкідливий код. Комунікаційні та обчислювальні можливості об'єктів інтелектуальних мереж є цілями для фізичного або віддаленого компрометації. Більш того, з тих пір як вони використовують різні типи програмного забезпечення, вони можуть бути метою різного виду зараження програмного забезпечення або зараження шкідливим кодом з метою контролю і маніпуляції ними. Також широко розгорнуті об'єкти з обмеженими пристроями (датчики і т.д.) зазвичай нестійкі до втручань, що робить фізичну компрометацію легким завданням;

– доступність і проблеми з DoS. У класичних енергомережах було важко, якщо не неможливо, вибрати метою атаки доступність активів (електричних лічильників, підстанцій і т.д.), особливо в великому масштабі. В інтелектуальних мережах, в яких кошти ІКТ інтегровані навіть в життєво важливі активи енергомережі, дозволяють вибрати їх як цілі, роблячи їх частково або повністю недоступними в результаті DoS-атаки. Більш того, беручи до уваги те, що більшість пристроїв (речей) підтримують протокол IP і не використовують власних протоколів, завдання зловмисника значно полегшується;

– інтелектуальна мережа може бути розглянута як найбільша кіберфізическая система, що включає фізичні системи подання фізичних активів інтелектуальної мережі (трансформатори, вимикачі, розумні лічильники, кабелі тощо) та ІКТ-системи, де ІКТ-елементи контролюють або управляють фізичними об'єктами [6].

Несанкціонований доступ до RFID. Несанкціонований доступ до міток, які можуть містити ідентифікаційну інформацію, – це головна проблема безпеки IoT (можливо розкриття будь-якої конфіденційної інформації про користувача), по-цьому вона повинна бути вирішена в першу чергу. Мітка може бути не тільки прочитана зчитувальних пристроїв зловмисника, але навіть модифікована або

пошкоджена. Існують кілька реальних загроз для RFID, які включають RFID-вірус, атаку за допомогою мобільного телефону і злом SpeedPass.

Порушення безпеки вузлів датчиків. WSN вразлива до деяких видів атак, по вузлам датчиків - це частина двосторонньої мережі датчиків, що означає не тільки можливість передачі, але і захоплення даних. Можливі атаки містять у собі забивання каналу, втручання, атаку Сібілли (мережева атака, при якій один з вузлів може мати кілька ідентифікаторів, тим самим порушуючи роботу системи), заповнення та деякі інші види атак, які полягають в наступному:

- 1) забивання ускладнює роботу всієї мережі, інтерферуючи з частотами вузлів датчиків;
- 2) втручання – це вид атаки, в якій інформація з вузлів може бути вилучена або змінена зловмисником для того, щоб взяти вузол під свій контроль;
- 3) атака Сібілли – нав'язування множинних псевдоанонімність ідентифікаційних даних, надаючи їм велике значення;
- 4) заповнення – це вид DOS-атаки, викликаний більший обсяг трафіку, що призводить до витрачання ресурсів пам'яті.

Зловживання хмарними обчисленнями. Хмарні обчислення – це велика мережа об'єднаних серверів, яка дозволяє ділитися ресурсами один з одним. Розділення ресурсів може зіткнутися з безліччю загроз безпеки, таких як Man-in-the-middle атака (MITM), фішинг і т.д. Альянс безпеки хмарних обчислень (CSA) покладав ще деякі загрози, такі як шкідливий інсайдер, втрати даних, крадіжка акаунтів, неймовірне використання комп'ютерів, які поділяють ресурси, які полягають в наступному:

- 1) шкідливий інсайдер – це загроза того, що легальний користувач, який має доступ до даних, може бути залучений в маніпуляції з даними;
- 2) втрата даних – це загроза, суть якої полягає в тому, що будь-який злочинець, що має несанкціонований доступ до мережі, може змінювати або видаляти існуючі дані;
- 3) Man-in-the-middle - це вид загрози крадіжки акаунтів, суть якої в тому, що атакуючий може змінювати або перехоплювати повідомлення в обміні між

					ВКБР-125.26.0028.00.00.ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		
						9

двома учасниками;

4) хмарні обчислення можуть бути використані жорстоким чином, оскільки якщо нападник отримує можливість завантажити будь-яке шкідливе програмне забезпечення на сервер, використовуючи, наприклад, ботнет, це може дати нападнику контроль над багатьма пов'язаними пристроями [2].

Маючи справу з алгоритмами безпеки, протоколами та методами інтелектуальних мереж, заснованими на Інтернеті Речей, необхідно взяти до уваги наступні завдання:

– масштабованість. Інтелектуальна мережа може сягати на величезній території (кілька міст або вся країна) і включати величезна кількість розумних пристроїв і об'єктів. Це робить важким розробку масштабованих рішень безпеки, таких, як управління ключами і автентифікація;

– мобільність. З мобільними пристроями (об'єктами), такими, як електрокари або польові технічні агенти, завжди буде необхідність автентифікації і безпечної комунікації із змінним оточенням (розумні лічильники, що заряджають станції і т.д.);

– розгортання. При охопленні всієї країни інтелектуальної мережею об'єкти (пристрої) розміщуються на величезних територіях, працюють без контролю і можуть бути поміщені у віддалені місця без будь-якого захисту по периметру, роблячи їх легкодоступними. Рішення безпеки повинні відслідковувати будь-які спроби втручання;

– успадковані системи. Раніше розміщені пристрої та системи можуть мати незначну або відсутню зовсім підтримку безпеки з тих пір, як вони стали ґрунтуватися здебільшого на власних рішеннях (апаратного або програмного забезпечення), розміщуватися на ізольованих островах без зв'язку або через приватну інфраструктуру зв'язку. Інтегрування цих успадкованих систем в інтелектуальну мережу, засновану на Інтернеті речей - це нетривіальне завдання, тим більше що в більшості випадків немає можливості замінити їх новими системами або оновити їх для підтримки бажаних рішень безпеки:

– обмежені ресурси. Деякі пристрої (об'єкти) інтелектуальної мережі,

особливо масово поширені і використовуються, обмежені в ресурсах. Особлива увага повинна бути приділена розробці рішень безпеки, щоб переконатися, що їх обмежені ресурси зможуть прийняти рішення;

– неоднорідність. Досягнення безпечної end-to-end-зв'язку для великої кількості видів ресурсів пристроїв (об'єктів), які використовуються інтелектуальною мережею (таких, як пам'ять, обчислювальна потужність, пропускна здатність, автономність і т.д.) і використовуваних ними протоколів і стеків зв'язку (для пристроїв, що не використовують IP-протокол) - це завдання, що вимагає більш частих адаптацій існуючих рішень або навіть використання шлюзів;

– сумісність, що розглядається як один із наслідків неоднорідності впроваджуваних протоколів і стеків зв'язку між пристроями (об'єктами) інтелектуальної мережі. Успадковані системи і пристрої (об'єкти), які не підтримують TCP / IP-стеки (такі, як, Zigbee v1, HART), не можуть зв'язуватися з системами, заснованими на IP, і пристроями (об'єктами) без використання шлюзів, роблячи end- to -end-безпечну зв'язок неможливою. Сумісність може бути розглянута між двома пристроями, що використовують однакові протоколи і стеки зв'язку, але різні можливості функції: одна - з повною під- Держкою, інша - з частковою (такий, як DTLS);

– початкова завантаження - це проблема ефективного виконання початкового завантаження мільйонів пристроїв (об'єктів) інтелектуальної мережі з необхідними початковими ключевими матеріалами (такими, як криптографічні ключі, криптографічні функції або алгоритми і параметри і т.д.);

– довірче управління. Об'єкти (пристрої) в інтелектуальній мережі можуть управлятися різними суб'єктами (кінцевий споживач для розумних приладів, оператор SG для розумних лічильників і датчиків і т.д.). Об'єкти (пристрої) не можуть зв'язуватися, якщо не встановлено мінімальний рівень довіри. У той час як об'єкти (пристрої), керовані одним підприємством, можуть легко встановити довірчі взаємини, побудова довіри між об'єктами (пристроями), що належать різним підприємствам, - це трудомістке завдання, особливо в великомасштабних

мережах;

– затримка/обмеження за часом. Деякі частини SG повинні відповідати в реальному часі на події та повідомлення. Наприклад, електричні SCADA-системи

(Supervisory Control and Data Acquisition), що використовуються на передавальних і розподільних підстанціях, повинні відповідати в реальному часі на будь-які зміни значень струму, напруги або частоти на додаток до інших метеорологічним параметрам, що впливає на функціонування обладнання, представленими різними розумними об'єктами (датчиками, приводами і т.д.), для того щоб забезпечити безпеку і запобігти поширенню аномалій (наприклад, перевантаження по потужності або перебою) на інші частини енергомережі. Це робить операції, які витрачають багато часу (тобто операції із загальним ключем), непридатними для використання [6].

Так у США у 2019 році був переданий законопроект, згідно з яким влада отримує повний доступ до пристроїв інтернету речей. Законопроект передбачає, що розробники таких пристроїв будуть передавати владі всю інформацію про прошивку, схемою роботи, будь-яких ризики злому пристроїв, а також способи протидії можливому злому. Зроблені за межами США пристрої, чиї розробники відмовляться надати відповідну інформацію, будуть заборонені до використання [33]. Також Великобританія запровадила законопроект щодо пристроїв IoT. Але він направлений скоріше не на моніторинг зі сторони влади, а на створення умов безпеки для користувачів [34]. Все це говорить про те що тема моніторингу та управління розвиватиметься і надалі, тому що безпека у державі стоїть на перших місцях і здатність контролювати пристрої IoT залишатиметься у пріоритеті.

Комфорт, зручність і безпека – це основна вимога до системи кібербезпеки комп'ютерної мережі розумного будинку. На сьогоднішній день існує велика кількість фірм які займаються системами кібербезпеки комп'ютерних мереж розумних будинків і пропонують низку своїх готових рішень. Розглянемо основні з них і проаналізуємо їх недоліки та переваги:

1. Ajax StarterKit – це українська фірма, яка пропонує свою систему

					ВКБР-125.26.0028.00.00.ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		
						12

моніторингу розумного будинку. В комплект системи входить:

- сирена;
- мобільний пульт керування;
- сенсори;
- датчики.

На рисунку 2.1 показано зовнішній вигляд системи фірми Ajax StarterKit:



Рисунок 2.1 - Система фірми Ajax StarterKit.

Перевагами системи є:

- захищений радіоканал;
- простота налаштування і керування;
- швидке оповіщення;
- резервне живлення контролера.

Недоліком ситеми є виключно охоронні функції.

2. Система Xiaomi Mijia яка поєднує в собі охоронні функції та функції моніторингу та керування побутовими пристроями розумного будинку. На рисунку 2.2 показано зовнішній вигляд складових системи фірми Xiaomi Mijia:



Рисунок 2.2 - Складові системи моніторингу фірми Xiaomi Mijia.

Перевагами системи є:

- реалізація функціоналу вже зі стартовим набором елементів;
- сумісність із сучасними мобільними платформами;
- обладнання надходить вже налаштованим.

Недоліком системи моніторингу Xiaomi Mijia є те, що китайські розетки несумісні з європейськими вилками і потребують додатково перехідника.

3. Система моніторингу розумного будинку фірми Redmond, Продукція цієї компанії відрізняється великим асортиментом виробів для систем моніторингу побутових пристроїв розумного будинку. На рисунку 2.3 показано зовнішній вигляд складових системи моніторингу розумного будинку фірми Redmond:



Рисунок 2.3 - Складові системи моніторингу фірми Redmond.

Перевагами системи моніторингу фірми Redmond є:

- можливість підключення великої кількості побутових пристроїв розумного будинку;
- реалізація повної автоматизації розумного будинку;
- надійність та якість системи моніторингу.

Недоліками системи є:

- відсутність голосового управління;
- дистанційне керування розумним будинком зі смартфона потребує наявності смарт-пристрою у самому будинку.

Як бачимо, на сьогоднішній день існує багато фірм які пропонують свої розробки для створення систем кібербезпеки комп'ютерної мережі розумного будинку. Але зважаючи на високу вартість таких систем мною було прийнято рішення створити власну систему кібербезпеки комп'ютерної мережі розумного будинку.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для побудови системи кібербезпеки комп'ютерної мережі розумного будинку використовуються:

- вбудовані системи – тут використовується як програмне забезпечення так і необхідне обладнання. Системи даного типу керують певними функціями, пов'язаними з більшою системою. Вбудовані системи створюються на основі мікроконтролерів або мікропроцесорів;
- інтелектуальні системи – такі пристрої побудовані на базі мікроконтролера і можуть виконувати певні обчислення;
- мікроконтролер (MCU) – ці непотужні комп'ютери мають ЦП, ПЗП і ОЗП. Вони мають більш обмежену потужність, ніж мікропроцесори і містять елементи, для виконання простих завдань;
- мікропроцесор (MPU) – функції ЦП можуть бути розподіленими між однією або кількома інтегральними мікросхемами (IC). Для виконання поставлених завдань мікропроцесор потребує наявності периферійних пристроїв, але витрати на обробку певних задач істотно зменшуються тому, що в них входить тільки вартість ЦП;
- пристрої, що непризначені для обчислень – дані пристрої лише встановлюють підключення і передають інформацію.
- фізичні пристрої-перетворювачі - перетворюють один вид енергії в інший. Якщо це стосується Інтернету речей, то до них відносять внутрішні сенсори і виконавчі пристрої, що передають дані пов мірі того, як об'єкти взаємодіють з їх середовищем [8, 10].

Для вибору необхідного обладнання та розуміння роботи системи кібербезпеки мережі розумного будинку маємо провести аналіз архітектури розумного будинку. Власне розумний будинок є IoT-системою, а говорячи про Інтернет речей, потрібно визначити його потенціал.

Для ініціювання будь-якої IoT-системи потрібно задіяти всі можливі (наявні) ресурси: програмне забезпечення, технологічне обладнання та інформаційні системи, якими б різними вони не були, в одну цілісну структуру, з метою формування інтегрованого, надійного та економічно ефективного рішення. Тобто

кожній проектованій IoT-системі потрібна міцна архітектура IoT, щоб мати спроможність виконувати проектований функціонал. А результуюча ефективність та придатність системи значною мірою буде залежати від якості розробленої інфраструктури [8, 10].

Не дивлячись на те, що кожна система IoT має свої особливості, основа кожної архітектури Інтернету речей і загального потоку процесів обробки даних мало чим відрізняється. Перш за все, вона складається з речей, які є об'єктами, підключеними до Інтернету, які за допомогою вбудованих датчиків і виконавчих механізмів здатні відчувати навколишнє середовище та збирати інформацію, яка потім передається шлюзам IoT. Наступний етап складається із систем збору даних IoT та шлюзів, збираючих величезну кількість необроблених даних, перетворюючи її у цифрові потоки, фільтрують та попередньо обробляють, для підготовки до аналізу. Третій шар представлений кінцевими пристроями, які відповідають за подальшу обробку та ретельний аналіз даних. На цьому рівні також можуть бути задіяними технології візуалізації та машинного навчання. Потім дані передаються в центри обробки, які можуть бути або хмарними, або локально встановленими. Саме тут дані зберігаються та аналізуються для отримання практичної інформації. Це чотири шари архітектури IoT, докладно описані на рисунку 2.4:



Рисунок. 2.4 - Основні складові IoT-архітектури

Як основа для кожної системи IoT, підключені пристрої відповідають за надання суті Інтернету речей, яка є даними. Для підбору фізичних параметрів у

зовнішньому світі або всередині самого об'єкта їм потрібні датчики. Вони можуть бути вбудовані в самі пристрої розумних будинків або реалізовані як самостійні об'єкти для вимірювання та збору даних.

Ще одним незамінним елементом цього шару є виконавчі механізми, щоможуть перетворювати дані, генеровані розумними об'єктами, у фізичні дії.

Важливим є також те, що об'єкти повинні мати можливість не тільки двосторонньо взаємодіяти з відповідними шлюзами або системами збору даних, але також мати можливість розпізнавати та розмовляти між собою для збору та обміну інформацією та співпраці в реальному часі, щоб використовувати значення всього розгортання. Особливо у випадку обмежених ресурсів та пристроїв, що працюють від акумуляторів, досягнення цього не є простим завданням, оскільки такий зв'язок вимагає великої обчислювальної потужності та споживає дорогоцінну енергію та пропускну здатність. Тобто, надійна архітектура може забезпечити ефективне управління пристроєм лише тоді, коли вона використовує спеціальні, безпечні та нескладні протоколи зв'язку, що є властивим багатьом IoT-системам. Апаратні складові системи кібербезпеки показано на рисунку 2.5:



Рисунок 2.5 - Апаратні складові системи кібербезпеки

Незважаючи на те, що цей рівень функціонує в безпосередній близькості від датчиків та виконавчих механізмів на певних пристроях, важливо описати

його як окремий етап архітектури IoT, оскільки він має вирішальне значення для процесів збору даних, фільтрації та передачі до крайової інфраструктури та хмарних платформ. З огляду на величезний обсяг введення та виведення даних, який може створити розгортання мільйонних пристроїв, можливості збору, відбору та транспортування даних повинні бути в центрі уваги. Як посередники між пов'язаними речами та хмарою та аналітикою, шлюзи та системи збору даних забезпечують необхідну точку зв'язку, яка пов'язує решту шарів (рис. 2.6).

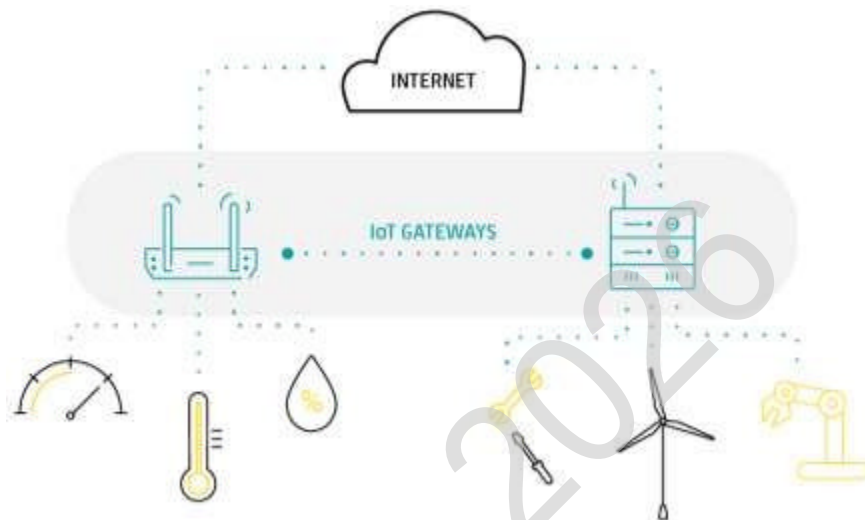


Рисунок 2.6 - Шлюзи та системи збору даних

Шлюзи полегшують зв'язок між датчиками та рештою системи, перетворюючи дані датчиків у формати, які легко передаються та використовуються для інших компонентів системи. Крім того, вони здатні контролювати, фільтрувати та відбирати дані, щоб мінімізувати обсяг інформації, яку потрібно пересилати в хмару, що позитивно впливає на витрати на передачу мережі та час відгуку. Таким чином, шлюзи забезпечують місце для локальної попередньої обробки даних датчиків, які формуються в корисні пакети і готові до подальшої обробки.

Ще одним аспектом, який підтримують шлюзи, є безпека. Оскільки шлюзи відповідають за управління інформаційним потоком в обох напрямках, за допомогою належних засобів шифрування та захисту вони можуть запобігти витоку хмарних даних IoT, а також зменшити ризик зловмисних зовнішніх атак на комп'ютерну мережу розумного будинку.

Є декілька простих порад, які можуть мінімізувати ризик зараження пристроїв комп'ютерної мережі розумного будинку:

- закрити доступ із зовнішньої мережі до пристрою без крайньої необхідності;
- періодичне перезавантаження допоможе позбутися від уже встановлених шкідників (але в більшості випадків ризик повторного зараження залишається);
- регулярна перевірка наявності нових версій прошивки та оновлення пристрою;
- використання складних паролей довжиною не менше 8 символів, що включають в себе букви різного регістра, цифри та спецсимволи;
- зміна заводських паролей після першого запуску пристрою, під час першого налаштування (навіть якщо пристрій про це не просить);
- закрити / заблокувати «зайві» порти, якщо є така можливість. Наприклад, якщо немає необхідності підключатись до роутера по Telnet (порт tcp:warto відключити його, щоб перекрити зловмисникам можливу лазівку.

Хоча ці поради й можуть частково закрити деякі існуючі проблеми та вразливості пристроїв мережі розумного, але це не вирішує проблему загалом, адже принцип розробки інтернету речей залишиться та нові вразливості будуть знайдені зловмисниками. Тож задача цієї роботи – це знайти можливість вирішити проблему безпеки комп'ютерної мережі розумного будинку системно.

2.3 Розгорнута постановка завдання

Кібербезпека комп'ютерної мережі розумного будинку починається зі збору даних шляхом контролю системи, з якою працює людина. У деяких випадках контролюються не самі пристрої, а зовнішнє середовище та процес, що

стосуються пристрою. В інших сценаріях може зацікавити моніторинг працездатності самих пристроїв окремо.

Показники апаратного забезпечення пристрою – це вимірювання самого обладнання або фізичного пристрою, як правило, з якимсь вбудованим датчиком. Такі вимірювання важливі для розуміння продуктивності апаратного забезпечення пристрою для задоволення його конструкторської функції, але можуть бути менш актуальними для тих, хто використовує пристрій. Ці показники можуть включати такі речі, як напруга акумулятора або потужність радіосигналу. Програмне забезпечення, що працює на пристроях, включає прикладне програмне забезпечення, а також саме системне програмне забезпечення, яке може бути операційною системою або шарами мережевого стека або драйверами пристроїв.

Вимірювання навколишнього середовища за допомогою датчиків часто є тим, про що люди думають стосовно пристроїв IoT. Очевидні показники тут – це ті, які можна безпосередньо виміряти за допомогою приєднаного датчика. Але датчики можуть також охоплювати локальний бездротовий зв'язок. Прикладами є звичні типи датчиків, такі як датчики для вимірювання температури або світла.

Рішення IoT - це складна система, що включає програмні компоненти, що працюють як на пристрої, так і в хмарі. Розуміння того, як взаємодіють ці дві системи, вимагає розуміння того, до якої інформації має доступ кожна сторона та як поєднати два середовища виконання програмного забезпечення.

У даному розділі проведений аналіз мережі розумного будинку з метою виявлення особливостей. Проаналізовано, які протоколи, технології та типи мереж застосовуються для побудови мережі, на якій архітектурі побудовані дані мережі, які пристрої беруть участь у роботі мережі розумного будинку.

Згідно з технічним завданням на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи кібербезпеки комп'ютерної мережі розумного будинку.

В процесі розробки випускної кваліфікаційної роботи за першим

(бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи кібербезпеки комп'ютерної мережі розумного будинку. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача;

д) сформулювати висновки про виконаний обсяг робіт та одержані результати.

КБПЗ – 2026

					ВКБР-125.26.0028.00.00.ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		22

3. ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Для забезпечення функціонування системи кібербезпеки комп'ютерної мережі розумного будинку потрібно вибрати таке апаратне забезпечення, яке забезпечуватиме виконання поставленої мети. Для цього пропонується наведене нижче обладнання.

Інтернет речей - це не тільки безліч різних приладів і датчиків, об'єднання між собою дротяними і бездротовими каналами зв'язку і підключати чинних до мережі Інтернет, а тісна інтеграція реального та віртуального світів, в середовищі якої спілкування здійснюється між людьми і пристроями.

Але як техніка розумного будинку може стати «розумною»? По-перше, за рахунок своєї конструкції - ця конструкція може бути такою, що поведінка системи буде виглядати розумною.

По-друге, за рахунок «інтелектуалізації» - оснащення системи моніторингу пристроями збору інформації, її обробки і прийняття рішень. Такий підхід дозволяє забезпечити досить складну і «розумну» поведінку набагато простішими способами, ніж за рахунок створення відповідної конструкції.

Компоненти системи кібербезпеки комп'ютерної мережі розумного будинку та навіть охоронні системи у значній кількості мають масові та грубі порушення принципів розробки:

- використання незмінних (hardcoded) і прихованих сервісних облікових даних;
- застосування однакових або легко передбачуваних ключів та ПІН-кодів;
- відсутність перевірки прав доступу при зверненні до відомої сторінки налаштувань (наприклад, /settings.asp в обхід /index.htm) або прямого виклику зображень і відео потоку IP-камери (на кшталт /axis-cgi/jpg/image.cgi);

– некоректна обробка отриманих даних, що викликає переповнення буфера. Як наслідок, можливе виконання довільного коду при отриманні зловмисно складеного TSP-пакета;

– примусове перемикання сервера на використання старих версій протоколів за запитом клієнтського пристрою;

– десятки інших типових помилок і навмисних послаблень безпеки заради зручності конфігурації нефахівцями (в тому числі віддаленого і без належної авторизації).

Уразливості наступних типів найбільш не захищені та для зловмисника дають більше гарантій успіху:

– виявлені після того, як виробник припинив підтримку пристрою і випуск патчів;

– виявлені недавно (для яких ще немає виправлень, або більшість користувачів не встигли виправлення застосувати);

– архітектурні недоліки, погано виправляються патчами ПЗ і рідко усуваються до кінця - на кшталт уразливості Spectre, існуючої в декількох різновидах і яка є актуальною на теперішній час;

– впливають відразу на кілька моделей і навіть типів пристроїв (наприклад, через загальне компонента веб-інтерфейсу або уразливості самого протоколу комунікації).

3.2 Розробка структурної схеми

Структурна схема системи кібербезпеки комп'ютерної мережі розумного будинку представлена на рисунку 3.1:

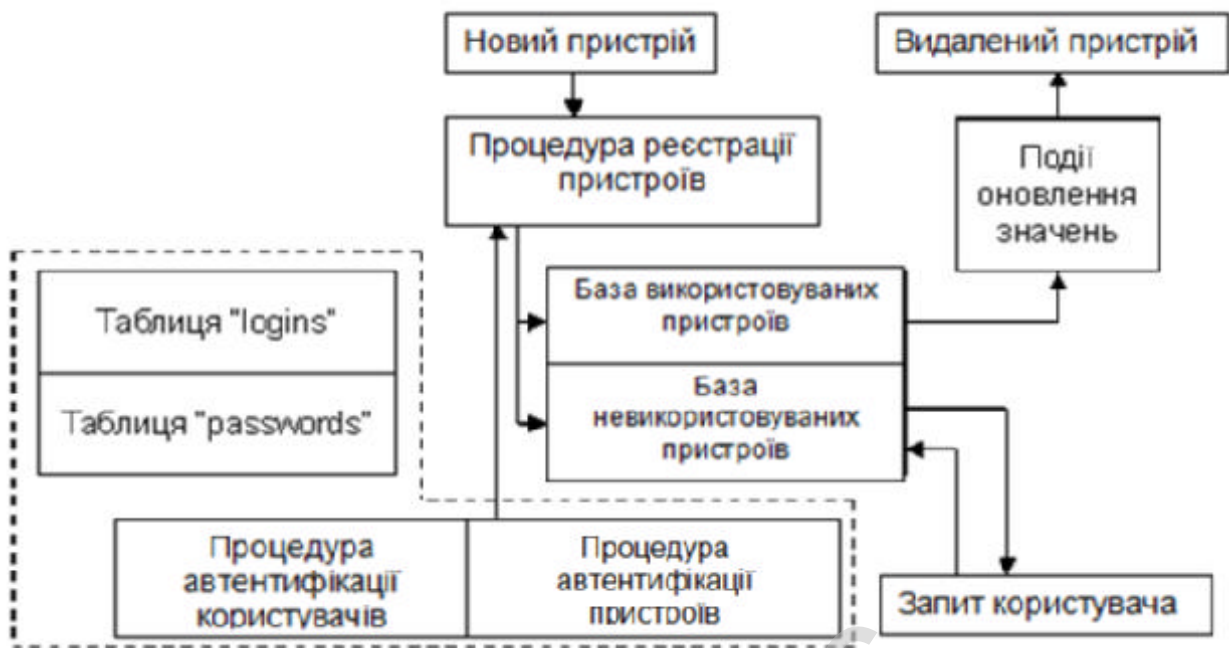


Рисунок 3.1 - Структурна схема системи кібербезпеки комп'ютерної мережі розумного будинку

Завдяки інтелекту і конективності обладнання отримаємо набір функцій, які можна розділити на чотири групи:

- моніторинг;
- управління;
- оптимізація;
- автономність.

Кожна функція, важлива і сама по собі, виявляється свого роду сходинкою для наступного рівня. Наприклад, функція моніторингу є основою для управління, оптимізації та автономності техніки. У результаті функція моніторингу дозволяє значно полегшити та оптимізувати користування системою, яка моніториться. Компанія може вибрати такий набір функцій, щоб її продукція була максимально корисною для споживача і таким чином, вона може зміцнювати свою конкурентну позицію.

З метою створення системи кібербезпеки комп'ютерної мережі розумного будинку та досягнення поставлених цілей пропонується міні-ПК Raspberry Pi. Він

здатний забезпечувати як і прості практичні завдання, наприклад вивчення комп'ютера і основ роботи з ним, інтернет-серфінгу, а також для програвання відео- та прослуховування аудіофайлів, так і більш складні завдання.

Raspberry Pi - це мініатюрний, розміром з кредитну карту, комп'ютер вартістю близько 25 доларів за базову модель і 35 - за більш новітню, який має величезну популярність і розійшовся по світу в кількості більше 4,5 мільйонів штук.

Однією з основних переваг міні-ПК Raspberry Pi є співвідношення якості продуктів і його вартості. В першу чергу, міні-ПК Raspberry Pi відводиться роль комп'ютера, призначеного для вивчення з його допомогою базових інформаційних технологій в школі. Позиціонується міні-ПК Raspberry Pi як дешеве рішення для початківців розробників. З урахуванням задовільною потужності цього пристрою, низького енергоспоживання і малої собівартості його можна використовувати для створення особистого міні сервера.

Стосовно до статистичного управління процесами – методу моніторингу виробничих процесів з метою управління якістю продукції безпосередньо в процесі виробництва, міні-ПК Raspberry Pi є першим доступним технічним рішенням такого розміру, яке можна використовувати повсюдно для програмування на багатьох мовах і в якості мікроконтролерів для управління роботизованими пристроями.

Одна з головних і привабливих особливостей міні-ПК Raspberry Pi - наявність на платі апаратних портів введення / виведення GPIO (General Purpose Input / Output, інтерфейс вводу / виводу загального призначення), що відкриває перспективи використання його в робототехнічних проектах і пристроях «розумного будинку» [14].

Також на ринку існують альтернативні пристрої по відношенню до міні-ПК Raspberry Pi. Такими альтернативами є Orange Pi Prime, Banana Pi M3, Rock64, ASUS Tinker board S, Libre Computer Renegade, Odroid H2, Arduino Mega 2560. Orange Pi Prime.

Відрізняється від Raspberry Pi 3 в кращу сторону наявністю 2 Гбайт ОЗУ і вбудованим в SoC AllWinner H5 відеоприскорювачем Mali-450 GPU, що дозволяє відтворювати 2K відео. Серед цікавих особливостей варто відзначити наявність ІК- приймача; платою можна управляти з пульта дистанційного керування або з деяких моделей стільникових телефонів з вбудованим ІЧ-світлодіодом. З нестандартного обладнання є також вбудований мікрофон і відеоінтерфейс CSI, що підтримує відеопотік до 1080p на швидкості 30 fps.

На платі розміром 98 × 60 мм знайшлося місце для роз'єму для карт пам'яті (до 32 Гбайт), Wi-Fi 802.11 b / g / n, Bluetooth 4.0, гігабітного Ethernet, чотирьох USB (три USB 2.0 Host і один USB 2.0 OTG) і сорокаконтактної GPIO-гребінки. Є навіть окремо виведений UART з TTL рівнями, так що ви можете в терміналі спостерігати за деталями завантаження Linux. З аудіо обладнання, крім згаданого вище мікрофона, є ще лінійний вихід і аудіовихід в HDMI. Відеоприскорювач підтримує OpenGL ES 2.0 і OpenVG 1.1. Серед підтримуваних ОС присутні Ubuntu, Debian і Android 5.1.

Banana Pi M3.

Флагман Banana Pi M3 побудований на базі восьмиядерного SoC Allwinner A83T (процесори ARM Cortex-A7, графічний процесор PowerVR SGX544MP1), розганяється до 1.8 ГГц і працює в оточенні 2 Гбайт ОЗУ і 8 Гбайт флеш-пам'яті. Крім гігабітного Ethernet, двох USB, Wi-Fi 802.11 b / g / n, Bluetooth 4.0 і HDMI, на платі присутній SATA. Так само, як і у Orange Pi Prime, у M3 є ІК-приймач, відеоінтерфейс CSI, оцінний UART, мікрофон, лінійний вихід і аудіовихід в HDMI. На відміну від Orange, у Banana є інтерфейс дисплея MIPI DSI, об'єднаний з I2C для сенсорного екрану. Природно, є і сорокаконтактна GPIO-гребінка.

Rock64.

Одноплатний комп'ютер Rock64 комплектується вже 4 Гбайт ОЗУ, що обслуговують 64-х бітний ARM Cortex A53, відеопідсистема здатна впоратися з потоком 4K на частоті 60 fps. Пристрій здатний харчуватися від POE. Графічна підсистема ARM Mali 450MP2 відповідає OpenGL ES 2.0, OpenVG1.1. На Rock64

					ВКБР-125.26.0028.00.00.ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		27

перенести ОС Debian, Cent OS, Fedora і Android 8, взагалі, потрібно відзначити, що розробники і ентузіасти цього комп'ютера портувала на нього велику кількість ОС, що базуються на Linux. У Rock64 ясна, детальна документація і живе, активне ком'юніті, так що, з урахуванням непоганих апаратних специфікацій і щадний ціни, цей одноплатнік - непоганий претендент на заміну Raspberry Pi 3 в проектах з підвищеними вимогами до «заліза».

Розробники Rock64 додали 64 контакти GPIO, вивівши на них навіть сигнали Ethernet, так що, якщо плануєте робити на одноплатніке управління розгалуженою зовнішньої периферією, придивіться до цього комп'ютера пильніше. Крім того, є USB3.0.

ASUS Tinker board S.Tinker побудований на базі SoC Rockchip RK3288 з чотирьох ядерним процесором ARM Cortex-A17 і працює під управлінням TinkerOS на базі Debian, можна встановити Android. Відеопроцесор Mali-T760 MP4 підтримує OpenGL ES 3.1, OpenCL 1.1, Renderscript і Direct3D 11.1.

Libre Computer Renegade.

Renegade, конструктивно розроблений настільки схожим на Raspberry, наскільки це тільки можливо; наприклад, є можливість розмістити ROC-Rk3328-CC прямо в корпусі Raspberry Pi.

SoC RK-3328 побудована на основі чотирьох ядерного процесора ARM Cortex-A53 з робочою частотою до 1.5 ГГц. SoC така ж, як і в Rock64, так що тут ви теж маєте той же GPU Mali 450MP2 з робочою частотою 500 МГц. При замовленні у вас є можливість варіювати обсяг ОЗУ, ви можете вибрати 1 Гбайт DDR4 (тоді комп'ютер обійдеться вам в \$ 35), 2 Гбайт за \$ 50 або 4 Гбайт за \$ 80. З операційних систем на даний момент доступні Ubuntu 18.04, Debian 9, OpenMediaVault 4, Station OS і Android 7.1 Odroid H2.

Odroid H2, побудований на базі 64-х бітного 4-х ядерного Intel Celeron Gemini Lake J4105, цілком може претендувати на шматок ринкової ніші, яку займає одноплатними комп'ютерами на базі процесорів ARM. Якщо вам миліше x86-сумісні рішення, можливо, ця невелика плата розміром 110 × 110 мм, з

					ВКБР-125.26.0028.00.00.ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		
						28

пасивним охолодженням, GPU Intel UHD Graphics 600, шиною PCI-E gen2 і здвоєним SATA 6 Гбайт / с доведеться вам до вподоби.

За специфікації Intel максимальний розмір ОЗУ дорівнює 8 Гбайт (SO-DIMM DDR4 2400 МГц, в комплект не входить, купується окремо, як і для будь-якого «Пісюк»), але виробник Odroid H2 стверджує, що встромляв дві планки по 16 Гбайт (разом 32 Гбайт) і все прекрасно працювало. Зрозуміло, плата підтримує Windows 10 / Linux x64, DirectX 12, OpenGL 4.3, OGL ES 3.0, OpenCL 2.0.

3.3 Розробка функціональної схеми

Для початку потрібно скласти список необхідного обладнання для системи кібербезпеки комп'ютерної мережі розумного будинку.

Підбір необхідного обладнання є важливим кроком, оскільки знадобиться не тільки сам центральний пристрій, такий як міні-ПК Raspberry Pi, а й периферійні. Для системи знадобиться сам центральний пристрій, в даному випадку міні-ПК Raspberry Pi. Для налаштування міні-ПК потрібно такі пристрої як клавіатура, маніпулятор типу миш, монітор, карта пам'яті.

Функціональна схема системи системи кібербезпеки комп'ютерної мережі розумного будинку представлена на рисунку 3.2:

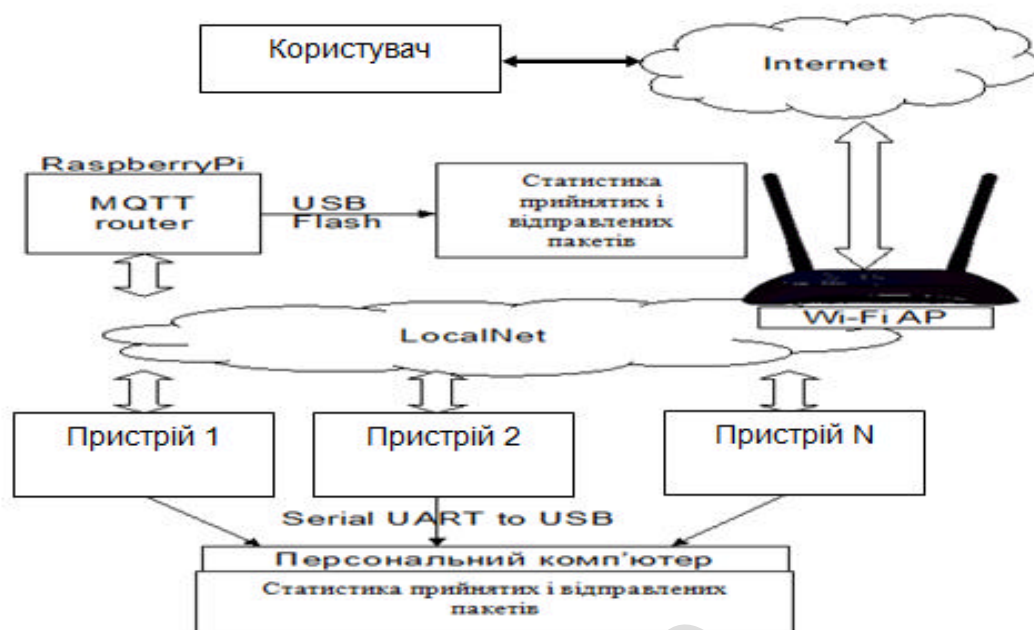


Рисунок 3.2 - Функціональна схема системи кібербезпеки комп'ютерної мережі розумного будинку

Повний список пристроїв необхідних для налаштування міні-ПК Raspberry Pi наведено нижче.

1. Одноплатний комп'ютер Raspberry Pi.
2. Монітор.
3. Клавіатура.
4. Маніпулятор типу «миша».
5. Адаптер живлення.
6. Карта пам'яті.

Також для роботи всієї системи необхідні датчики. Вони є невід'ємною частиною даної системи. Повний список необхідних датчиків наведено нижче.

1. Датчик температури DS18B20.
2. Датчик температури та атмосферного тиску BPM180.
3. Датчик освітленості TSL2561.

Так як міні-ПК Raspberry Pi не має у своєму Wi-Fi модулю, потрібно встановити Wi-Fi адаптер, який підключається шляхом підключення до порту

USB, а саме EDUP Nano EP N8531. Адаптер Wi-Fi дозволяє підвищити мобільність даної системи і не бути прив'язаним до мережевого кабелю.

У таблиці 3.1 наведений повний перелік пристроїв, їх моделі та необхідна кількість.

Таблиця 3.1 - Перелік додаткового обладнання

Тип пристрою	Модель	Кількість
Монітор	Samsung S22R350FHI	1
Клавіатура	Logitech Keyboard K120	1
Маніпулятор	Logitech B110	1
Wi-Fi адаптер	EDUP Nano EP N8531	1
Адаптер живлення	Grand-X CH-65B	1
Карта пам'яті	Transcend Premium microSDHC 16GB Class 10 UHS-I	1
Датчик температури	DS18B20	2
Датчик температури та атмосферного тиску	BPM180	1
Датчик освітленості	TSL2561	1

Перші версії міні-ПК Raspberry Pi: моделі «А», «В» (рис. 2.1), «А+» і «В+» - оснащені процесором Broadcom BCM2835 архітектури ARM 11 з тактовою частотою 700 МГц і модулем оперативної пам'яті на 256 (або 512) Мбайт, розміщеним за технологією package-on-package безпосередньо на процесорі. Модель «А» оснащена одним портом USB 2.0, модель «В» – двома такими портами, модель «В+» – чотирма. У моделей «В» і «В+» також є в наявності порт Ethernet. Крім основного ядра, процесор BCM2835 має у своєму складі графічне ядро з підтримкою OpenGL ES 2.0, апаратного прискорення та відео Full HD, а також ядро DSP (Digital Signal Processor, цифровий сигнальний процесор).

Комп'ютер працює через роз'єм MicroUSB, при цьому сила струму повинна становити мінімум 0,5-0,7А. При менших значеннях комп'ютер все ще може включитися, але буде йти в перезавантаження при запусці ресурсоємних задач. Отже, підключати плату краще не через хаб, а безпосередньо до USB порту комп'ютера або в розетку через спеціальний перехідник.

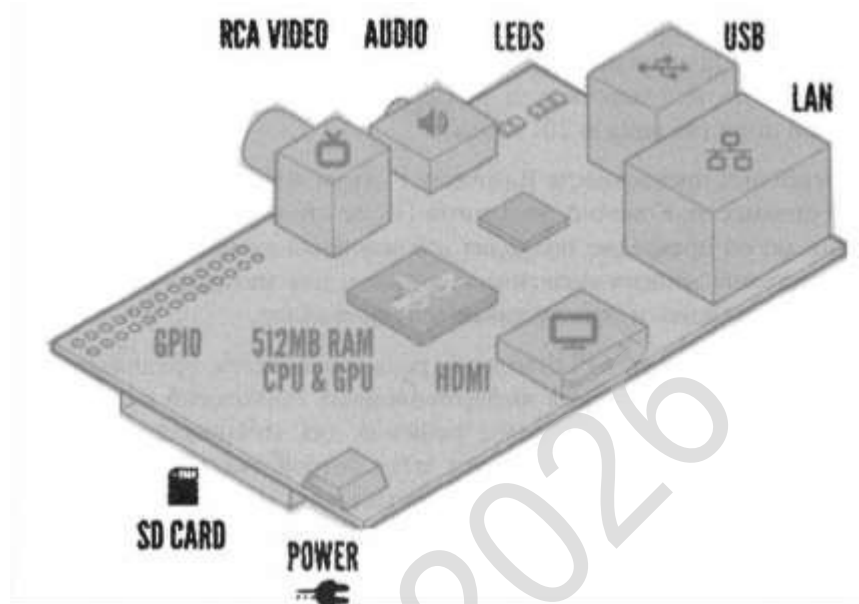


Рисунок 3.3 - Схема міні-ПК Raspberry Pi, Модель «В»

Ніяких кнопок ввімкнення/вимкнення на платі міні-ПК Raspberry Pi немає. Якщо необхідно пристрій запустити – потрібно підключити USB-живлення, для вимкнення - висмикнути шнур. Залишається сподіватися, що в майбутніх ревізіях міні-ПК Raspberry Pi, можливо, добавлять живлення по Ethernet, оскільки це один з найбільш частих запитів від користувачів.

Версії «А» і «В» містять слот для карт пам'яті формату SD, версії «А +» і «В +» - слотом для карт формату MicroSD. Без карти пам'яті міні-ПК Raspberry Pi не включається, оскільки саме на ній повинна бути записана операційна система, - це все одно що намагатися запустити комп'ютер без жорсткого диска. Оскільки власної ОС у внутрішній пам'яті (як, наприклад, в телефонах) у міні-ПК Raspberry Pi немає, то з цього випливає один позитивний момент - пристрій практично неможливо перетворити в «цеглина»: після будь-якого невдалого експерименту

досить перезаписати дистрибутив на карті пам'яті, і міні-ПК Raspberry Pi знову запрацює як новенький.

Для підключення дисплея на платі міні-ПК Raspberry Pi є відразу два інтерфейси: RCA Video (композитний) і HDMI. Застосовуючи відповідні перехідники, можна вийти і на більш традиційні: VGA і DVI. HDMI підтримує передачу як відео, так й звуку, але якщо буде потрібно окремий аудіо-канал, то і він присутній на платі у вигляді стандартного міні-джека 3,5 мм. Підключення мікрофона також можливо, але для цього знадобиться знайти USB-пристрій, сумісний з міні-ПК Raspberry Pi.

Існуючі зараз моделі міні-ПК Raspberry Pi не мають модуля Wi-Fi, і для роботи з ними в Інтернеті знадобиться задіяти порт Ethernet. Оскільки фізично на платі він скомутований через USB 2.0, то забезпечує не гігабітну, а всього лише 100-мегабітну швидкість.

Чіпи процесора і графічного прискорювача не оснащені навіть найпростішими радіаторами, і після декількох годин роботи комп'ютера стає очевидним, чому його розробники зупинилися саме на цьому рішенні: плата нагрівається при роботі зовсім незначно - вона швидше тепла, ніж гаряча.

Частота процесора, як уже згадувалося, становить 700 МГц, і, в залежності від дистриб'ютора, його можна розігнати до 1000 МГц без втрати гарантії (можливий вибір і більш сприятливих режимів). Чіп пам'яті виробництва Samsung або Нупіх напаяний прямо поверх основного чіпсета, так що збільшити RAM самостійно не вийде. При покупці варто звернути увагу на маркування цієї SoC (System-on-a-Chip, системи на кристалі): номер партії для «старих» версій моделі «В» з 256 Мбайт RAM починається з K4P2G, а у випуску з 512 Мбайт пам'яті - з K4P4G.

Відеоприскорювач Broadcom VideoCore N дозволяє навіть при такому слабкому процесорі декодувати відео 1080p h.264 з бітрейтом аж до 40 Мбіт / с. Для апаратного прискорення MPEG-2 і VC-1 ліцензії доведеться докуповувати окремо.

					ВКБР-125.26.0028.00.00.ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		
						33

Плата оснащена індикаторами - п'ятьма світлодіодами: три з них демонструють активність і режим роботи Ethernet, а ще два сигналізують про наявність живлення і роботі з SD-картою.

А тепер перерахуємо набір низькорівневих інтерфейсів:

- по-перше, на платі є слоти CSI-2 для підключення камери і DSI – для підключення дисплею;

- по-друге. присутня колодка на 26 (40 для версії «B+») ліній вводу та виводу загального призначення GPIO. На них же реалізовані інтерфейси UART, консольний порт, шина SPI (Serial Peripheral Interface, послідовний периферійний інтерфейс), PC (Inter-Integrated Circuit, послідовна шина даних для зв'язку інтегральних схем) і I2S (Integrated Inter-chip Sound, послідовна шина даних, що служить для з'єднання цифрових аудіопристроїв).

Втім, недоліків у міні-ПК Raspberry Pi теж вистачає. У ньому, наприклад, немає власних годин реального часу (Real Time Clock, RTC), тому єдиний спосіб отримання часу – це синхронізація з NTP-серверами. SoC містить в собі цифровий сигнальний процесор (DSP), але повного доступу до його API досі немає.

У новій версії – міні-ПК Raspberry Pi 2 Model B (рис. 3.2), замість процесора BCM2835 в ній встановлений чіп BCM2836. Відрізняється він від попередника наявністю чотирьох ядер ARM Cortex-A53 з набором інструкцій ARMv7 - проти ARMv6k у BCM2835. Тактова частота цього процесора складає 1400 МГц – не надто багато за мірками сучасних рішень Qualcomm і навіть MediaTek, але для переважної більшості DIY-проектів (Do It Yourself, «Зроби сам») її вистачить з лишком, тим більше що, нагадаємо, ядер на цей раз цілих чотири. Об'єм оперативної пам'яті нового пристрою збільшений з 512 Мбайт до 1 Гбайт. Відеоядро змін не зазнало - це як і раніше Broadcom VideoCore IV. Розробники прогнозують шестиразовий приріст продуктивності в багатопоточних тестах і триразовий в однопоточних [14].

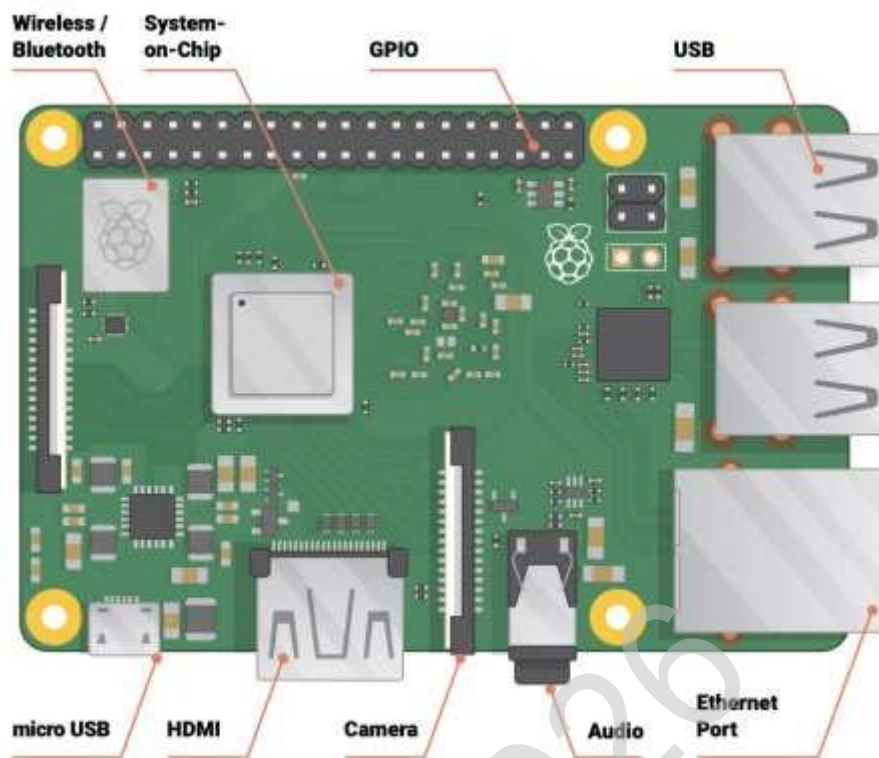


Рисунок 3.4 - Схема міні-ПК Raspberry Pi 2 Model B

Додаткове обладнання

Характеристики монітора:

- діагональ дисплею – 22 дюйм;
- частота оновлення – 60 гц;
- максимальна роздільна здатність – 1920 * 1080;
- тип матриці – tn;
- яскрість дисплею – 250 кд/м²;
- інтерфейси – VGA, HDMI;
- контрастність – 1000:1;
- співвідношення сторін – 16:9;
- максимальна кількість кольорів – 16,7 млн. кольорів;
- споживна потужність:

(а) напруга: 100 ~ 240в;

(b) енергоспоживання:

1. робочий режим: 23Вт;
2. Режим очікування: 0,3Вт. Клавіатура:
 - кількість клавіш – 104;
 - тип клавіатури – мембранні;
 - інтерфейс – USB.

Маніпулятор типу «миша» :

- джерело живлення – USB кабель;
- довжина кабелю – 1,8 м.;
- сумісність з ОС – ОС Windows;
- тип сенсору – оптичний.

Wi-Fi модуль:

- частота – Wi-Fi 2412-2484 МГц;
- стандарт – 802.11 b/g/n;
- потужність – +20дБ;
- типи шифрування – WEP, WPA, WPA2;
- режими роботи – клієнт (STA), точка доступу (AP), клієнт + точка доступу (STA+AP);
- напруга живлення – 1,7-3,6 В;
- споживання току – 70 мА (пиковое значение 240 мА);
- температурний режим – від -40 до +70 °С. Адаптер живлення:
- вхідна напруга – 220 В;
- вихідна напруга – 5 В;
- вихідний струм – 3.1 А;
- розмір – 88 x 45 x 25 мм.;
- вага – 73 г.;
- довжина кабелю – 114 см.

Карта пам'яті:

- розміри – 11 mm x 15 mm x 1 mm;
- форм-фактор – microSD карта пам'яті;

- тип Flash пам'яті:
 - 1. Пам'ять типу MLC;
 - 2. Пам'ять типу 3D;
 - ємність – 8 ГБ / 16 ГБ / 32 ГБ / 64 ГБ / 128 ГБ;
 - робоча напруга – 2.7В ~ 3.6В;
 - тест на падіння – падіння з 1.5 метрів;
 - робоча температура – стандартний діапазон-25 ° C (-13 ° F) ~ 85 ° C (185 ° F);
 - швидкість читання – (макс.) До 90 Мб / с.
- Датчик температури DS18B20:
- розрядність – 9-12 біт;
 - час перетворення – 750nS(max);
 - точність вимірювання $\pm 0.5\%$ в області температур – -10+85°C;
 - напруга живлення для точності вимірювання $\pm 0.5\%$ – 3,0-5,5V. Датчик температури та атмосферного тиску BPM180:
 - діапазон вимірювань: 300 - 1100 гПа (9000 ... -500 метрів над рівнем моря);
 - діапазон виміру температур: -40 to + 85 ° C (точність ± 2 градуси);
 - напруги живлення: 2 - 5В;
 - споживання в режимі очікування: 0,1 мкА. Датчик освітленості TSL2561:
 - робоча напруга живлення - 3.3-5 В;
 - чутливість – 65536 градацій;
 - 2 вбудованих інтегруючих модуля АЦП;
 - шина даних – I2C;
 - освітленість – від 0.1 до 40000 люкс;
 - точність – 0.1 люкс;
 - розміри – 14 мм на 18 мм.
- Wi-Fi адаптер EDUP Nano EP N8531:
- мережевий стандарт – IEEE 802.11n, IEEE 802.11g (draft), IEEE 802.11b;
 - частотний діапазон – 2.4-2.4835GHz;
 - робоча зона – до 100 м в приміщенні / до 200 м на відкритому просторі

швидкість передачі даних до 150Mbps;

– число каналів – 13;

– інтерфейс – USB.

Але, з точки зору практики, можливий такий варіант, що для налаштування міні-ПК Raspberry Pi вже є у наявності периферійні пристрої. Тому загальна вартість системи може значно скоротитись. Таким чином, якщо не брати до уваги периферійні пристрої, вартість системи становить 1910 гривень [16-20].

Так як міні-ПК Raspberry Pi не має вбудованої внутрішньої пам'яті, потрібно попередньо підготувати її. Для цього було обрано карту пам'яті формату MicroSD Transcend Premium microSDHC 16GB Class 10 UHS-I. Вона має у своєму складі 16 гб постійної пам'яті, чого вистачить для встановлення системи та подальших маніпуцій. Щоб встановити карту потрібно посунути її у слот MicroSD (рис. 3.5).

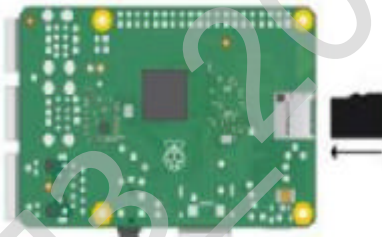


Рисунок 3.5 - Підключення карти пам'яті до міні-ПК Raspberry Pi

Потім потрібно підключити засоби введення та виведення. Для початку підключимо клавіатуру та маніпулятор типу «мишь». Для таких задач Raspberry Pi має у своєму складі usb порти, у які можна підключити периферію.

Наступним кроком є підключення монітору. Для цього потрібно з'єднати Raspberry Pi та монітор HDMI кабелем та підключити відповідно у HDMI порти.

У загальному вигляді маємо пристрій, до якого підключені периферія по кабелю USB (білий колір кабелю), монітор по кабелю HDMI (червоний колір кабелю), а також адаптер живлення через micro USB (чорний колір кабелю) (рис.3.6):

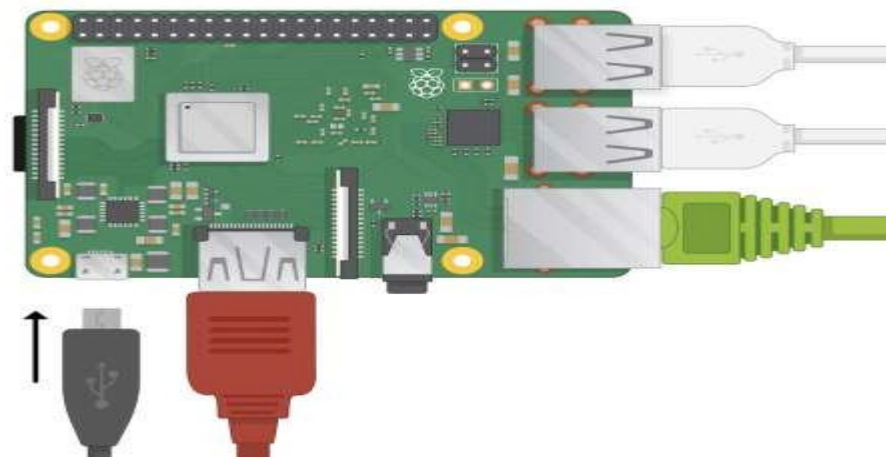


Рисунок 3.6 - Загальний вигляд міні-ПК Raspberry Pi з підключеними пристроями

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання бакалаврського проектування, наведена на рисунку 3.7:

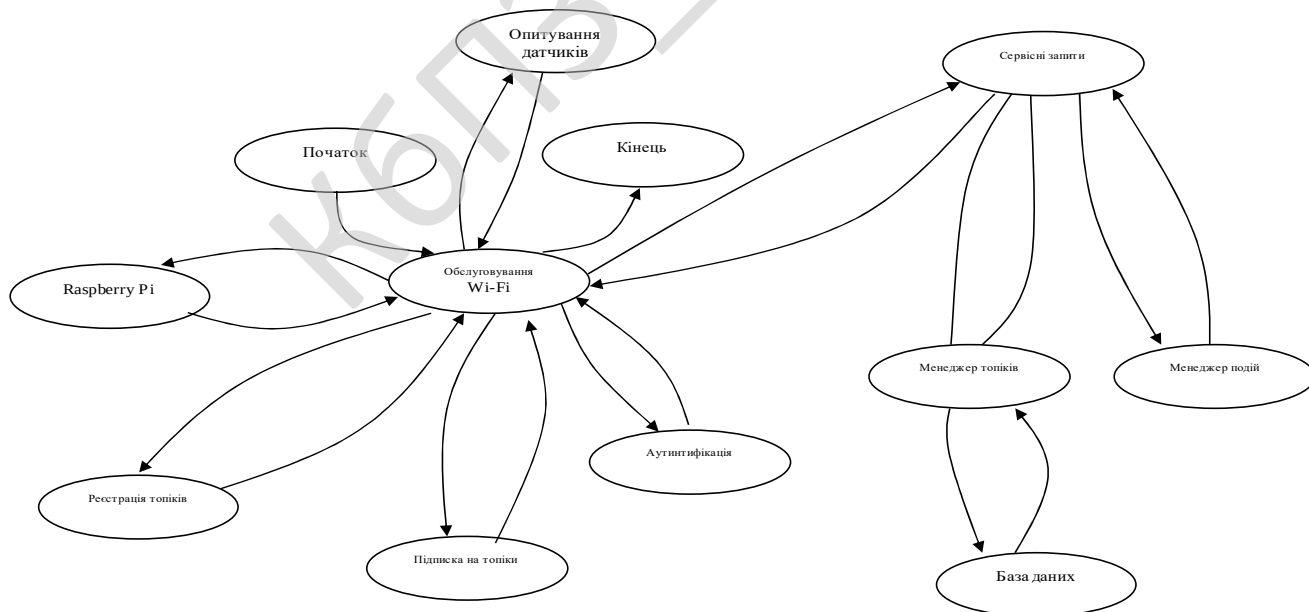


Рисунок 3.7 - Діаграма взаємодії процесів

У результаті узагальнена схема системи кібербезпеки комп'ютерної мережі розумного будинку на основі міні-ПК Raspberry Pi виглядатиме як на рисунку 3.7:

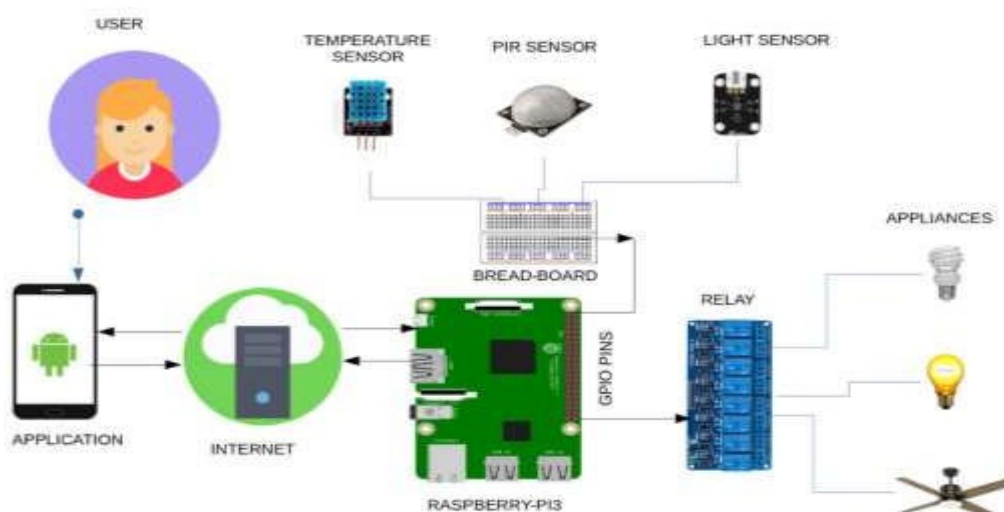


Рисунок 3.8 - Загальна схема системи кібербезпеки комп'ютерної мережі розумного будинку

До міні-ПК Raspberry Pi також мають здатність підключатися необхідні для роботи системи датчики, а також є можливість підключатися до мережі інтернет [21].

У даному розділі було створено функціональну і структурну системи кібербезпеки комп'ютерної мережі розумного будинку, а також діаграму процесів.

Для забезпечення функціонування системи системи кібербезпеки комп'ютерної мережі розумного будинку розумного будинку потрібно вибрати таке апаратне забезпечення, яке забезпечуватиме виконання поставлених задач та цілей. Тому було запропоновано міні-ПК Raspberry Pi, як «мозок» системи. Він здатний забезпечувати як і прості практичні завдання, наприклад вивчення комп'ютера і основ роботи з ним, а також для програвання відео- та прослуховування аудіофайлів, так і більш складні задачі, наприклад бути частиною більш складних систем. Міні-ПК Raspberry Pi є найбільш прийнятним варіантом, з економічної точки зору для побудови системи кібербезпеки комп'ютерної мережі розумного будинку розумного будинку.

4. РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Розробка блок-схем та опис алгоритмів функціонування системи

Перш ніж розпочати використання міні-ПК Raspberry Pi, потрібно налаштувати його програмне забезпечення, зокрема його операційну систему, програмне забезпечення, яке контролює, що може робити Pi. NOOBS, нове готове програмне забезпечення, покликане максимально спростити це, дозволяючи вибрати один із декількох різних операційних систем і встановити їх автоматично. NOOBS можна скачати з сайту Raspberry та встановити в карту пам'яті.

Завантажений образ операційної системи необхідно записати на карту пам'яті microSD. Це можна зробити за допомогою утиліти Win32DiskImager.exe.

Коли міні-ПК Raspberry Pi вперше увімкнеться або завантажиться зі свіжою інсталяцією NOOBS на своїй картці microSD, з'явиться екран з логотипом міні-ПК Raspberry Pi і невелике вікно прогресу у верхньому лівому куті.

Меню NOOBS – система, яка дозволяє вибрати операційну систему для запуску на міні-ПК Raspberry Pi. У стандартну комплектацію NOOBS входять дві операційні системи: Raspbian, версія операційної системи Debian Linux, спеціально розроблена для міні-ПК Raspberry Pi та LibreELEC, версія програмного забезпечення Kodi Entertainment Center. Якщо міні-ПК Raspberry Pi підключений до мережі - або за допомогою дротового з'єднання, або за допомогою опції „Мережі Wi-Fi (w)“ у верхній панелі піктограм - також можливо завантажити та встановити інші операційні системи.

На рисунку 4.1 показано блок-схему роботи системи:

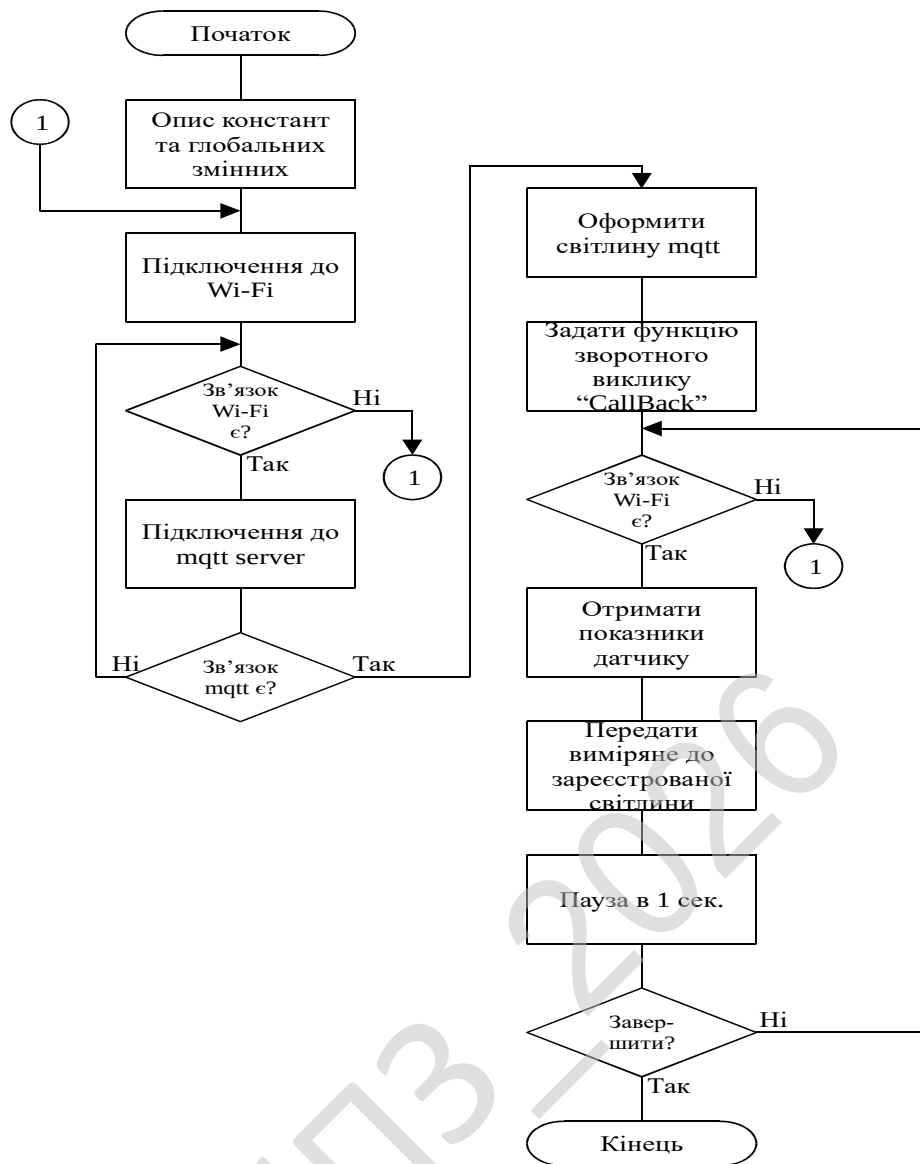


Рисунок 4.1 - Блок-схема роботи системи

На рисунку 4.2 показано блок-схему роботи серверної частини:

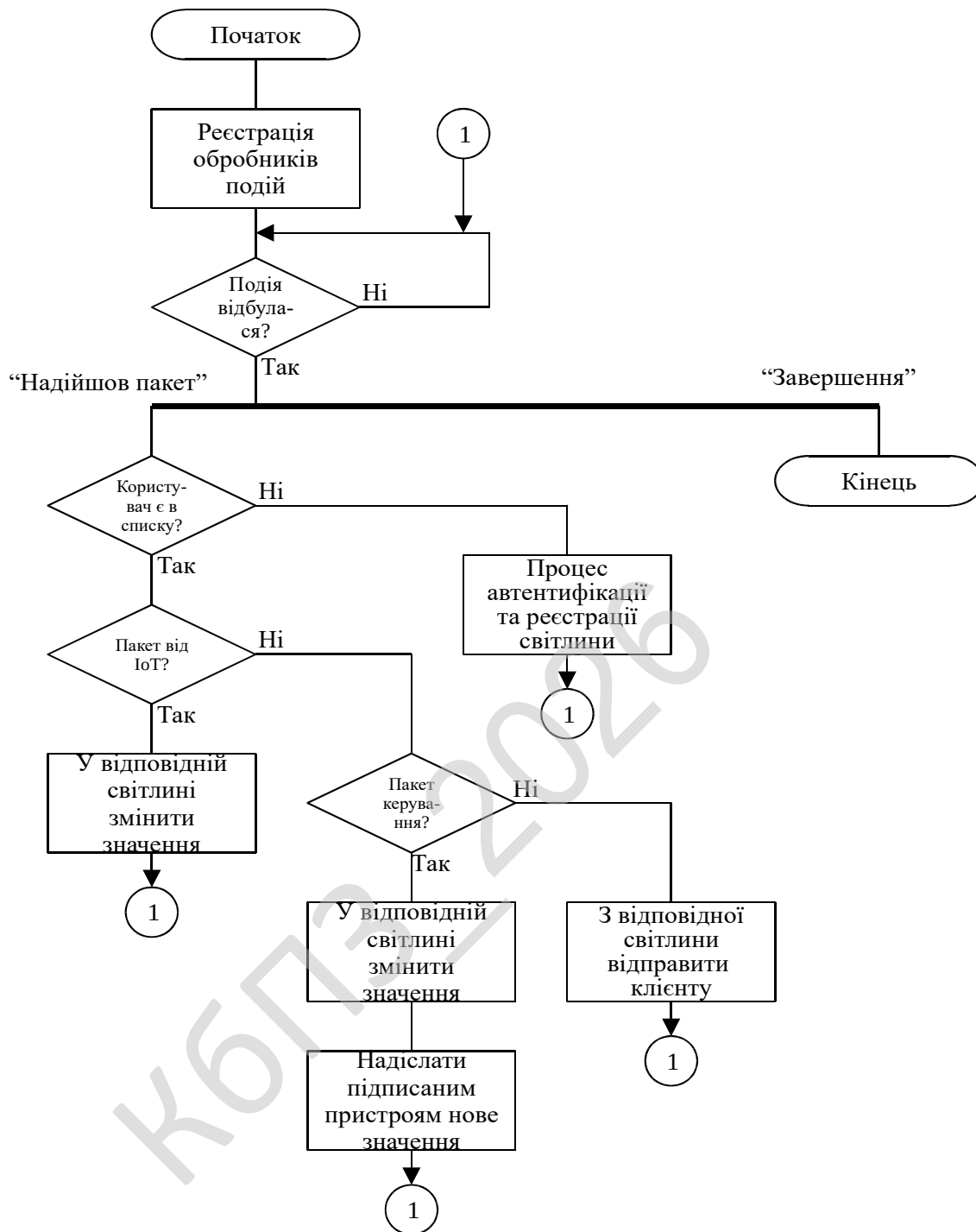


Рисунок 4.2 - Блок-схема роботи серверної частини

На рисунку 4.3 показано блок-схему роботи функції отримання повідомлень від сервера:



Рисунок 4.3 - Блок-схема роботи функції отримання повідомлень від сервера

Після встановлення потрібно перезавантажити міні-ПК Raspberry Pi. Далі систему буде встановлено.

Після включення користувач одразу побачить робочий стіл та майстер налаштувань операційної системи [21, 22].

Швидше за все, міні-ПК Raspberry Pi не використовуватиметься для виведення інформації на монітор - віддалений доступ до нього по ssh, тому зараз слід налаштувати його параметри мережі.

Встановлення мережевого з'єднання для міні-ПК Raspberry Pi забезпечує файл конфігурації / etc / network / interfaces, і якщо IP-адреса міні-ПК Raspberry Pi отримує по DHCP, то нічого правити в цьому файлі вам не доведеться. У разі ж призначення міні-ПК Raspberry Pi статичного IP-адреси, виконуємо команду:

```
sudo nano /etc/network/interfaces
```

і дописуємо в файл конфігурації / etc / network / interfaces наступні рядки:

```
iface eth0 inet static address 192.168.0.117
netmask 255.255.255.0
gateway 192.168.0.28 auto eth0
де:
```

iface eth0 inet static - вказує системі, що інтерфейс (iface eth0) знаходиться в діапазоні адрес IPv4 (inet) зі статичним IP-адресою (static);

– address 192.168.0.117- вказує, що IP-адреса (address) нашої мережевої

карти 192.168.0.117;

- netmask 255.255.255.0 - вказує, що наша маска підмережі (netmask) має значення 255.255.255.0;
- gateway 192.168.0.28 - вказує, що адреса шлюзу (gateway) за замовчуванням 192.168.0.1;
- auto eth0 - вказує, що інтерфейс eth0 необхідно включати автоматично при завантаженні системи з зазначеними тут параметрами.

Якщо потрібно змінити адреси серверів DNS, виконуємо команду:

```
sudo gedit /etc/resolv.conf
```

і вписуємо в розпочатому файлі конфігурації resolv.conf наступні рядки:

```
nameserver 192.168.1.1
```

```
nameserver 8.8.8.8
```

де: 192.168.1.1 і 8.8.8.8 - адреси серверів DNS. Якщо потрібно додати більше адрес - кожен адресу слід починати з нового рядка і з слова nameserver [14].

Далеко не завжди зручно вести до міні-ПК Raspberry Pi мережевий провід. Вирішити цю проблему можна за допомогою адаптера Wi-Fi, що підключається через USB. У попередньому розділі було зазначено, що цим адаптером є EDUP Nano EP N8531.

Далі підключаємо адаптер до USB – порту та перевіряємо, чи визначить його система. Для цього виконуємо команду iwconfig і знаходимо адаптер.

Потім шукаємо доступну бездротову мережу командою:

```
sudo iwlist wlan0 scan
```

SSH - це протокол, який забезпечує з'єднання по мережі (локальної або інтернет) між двома віддаленими пристроями. По ньому можна передавати майже будь-яку інформацію:

- файли;
- відео та аудіо в потоці;
- команди.

У нього є два значимих переваги. По-перше, він вміє на льоту стискати дані. По-друге, він створює зашифрований тунель.

Перше, що передбачає на міні-ПК Raspberry Pi SSH налаштування - запуск міні-ПК Raspberry Pi і відкриття терміналу. У консолі потрібно виконати команду `raspi-config`. Через мить після натиснення на Enter відобразиться інтерфейс стандартної утиліти конфігурації Raspberry. У ній необхідно знайти пункт **Interfacing Option** (рис. 4.4).

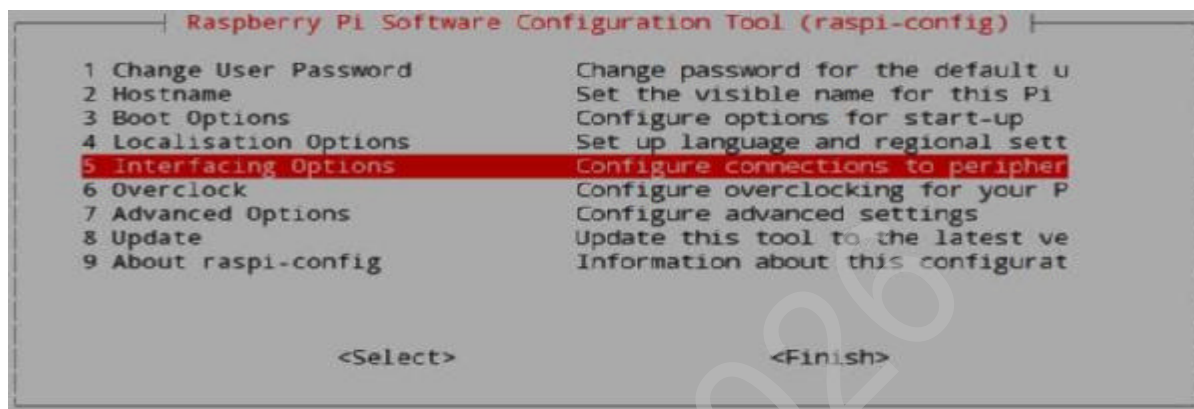


Рисунок 4.4 - Утиліта конфігурації Raspberry

Тепер потрібно зробити на міні-ПК Raspberry Pi enable SSH. Для цього слід виділити пункт SSH, натиснути Enter і вибрати Enable, а потім знову натиснути на "Введення". Потім залишиться повернутися на головний екран і вибрати Finish. Після цього всі внесені зміни будуть застосовані.

Тепер слід розглянути питання щодо того, як виконується підключення по SSH до Raspberry Pi. Розглянемо варіант з'єднання на платформі Windows.

Перше, що знадобиться - спеціальний додаток, так як в Windows немає своєї програми для роботи по протоколу Secure Shell. Найпопулярнішим з них є PuTTY - це повністю безкоштовний SSH-клієнт.

У вікні налаштування PuTTY (рис. 4.5) на вкладці "Сеанс" потрібно зробити наступне:

- вказати адресу Raspberry в поле "Ім'я хоста";
- ввести номер порту (для SSH найчастіше використовується 22-й, і він

буде саме таким, якщо користувач сам нічого не змінив);

- в списку перемикачів "Тип з'єднання" повинен бути активний SSH.

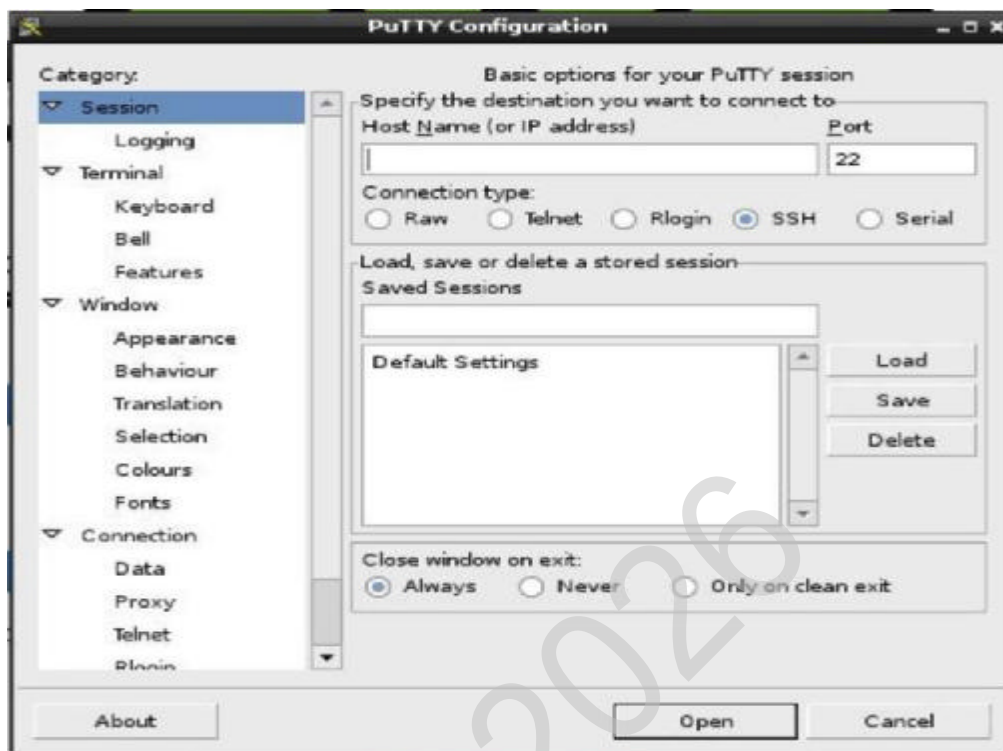


Рисунок 4.5 - Вікно налаштування PuTTY

Коли всі попередні дії зроблені, потрібно натиснути на кнопку «з'єднання». Якщо адреса, порт і тип з'єднання були вказані правильно, а мережа на двох пристроях працює справно, то через мить з'явиться запрошення командного рядка Windows, і через неї можна буде взаємодіяти з міні-ПК Raspberry Pi і стануть доступні всі команди.

Коли з'явиться консоль, потрібно пройти авторизацію в системі. Для цього потрібно, по-перше, вказати логін (ім'я користувача), а, по-друге, пароль, який призначений для відповідного користувача.

Безпека комп'ютерної мережі розумного будинку. Потрібно відійти від відкритих портів за замовчуванням до закритих за замовчуванням і прийняти передові практики посилення безпеки. Пристрої мережі повинні приймати до уваги стандартні мережеві конфігурації, які обмежують доступ до віддаленої

адреси для цих пристроїв локальним мережам або окремим постачальникам. Окрім забезпечення мережевої безпеки, необхідно застосувати ASLR, межі розмежування. З точки зору відповідності, сертифікація може допомогти споживачам отримати більш безпечний вибір, а також натиснути на виробників для створення ними більш безпечних продуктів.

Автоматичні оновлення. Автоматичні оновлення, є стандартом для настільних і мобільних операційних систем, вони надають розробникам механізм своєчасного для виправлення помилок та вразливостей без обтяження споживачів технічними завданнями або вимаганнями відклику. Для автоматичних оновлень потрібна модульна архітектура програмного забезпечення, за допомогою якої можна безпечно перезаписати основні модулі з можливістю відкату у разі виходу з ладу. Вони також вимагають криптографічних примітивів для пристроїв з обмеженим ресурсом та створення інфраструктури PKI для підтримки надійних оновлень. Окрім цих викликів, оновлення також вимагає, щоб спільнота IoT активно відстежувала в себе вразливі місця, що є потенційно обтяжливою відповідальністю з огляду на різноманітність пристроїв. Нагорода за пошук помилок може допомогти у цьому відношенні: приблизно 25% усіх вразливостей, які було виправлено в 2015 році в браузерах Chrome та Firefox, отримано за допомогою цієї системи. А Netgear тільки в 2017 році запустив банер для помилок для ПЗ у своєму маршрутизаторі. У випадку експлуатації з нульовим днем, яка вимикає автоматичні оновлення – розробники IoT повинні забезпечити безпечний механізм резервного копіювання, який, ймовірно, потребує фізичного доступу та втручання споживачів.

Зараження Deutsche Telekom та подальші виправлення дають чудовий приклад цього питання. Маршрутизатори DT мали вразливість, яка дозволила розповсюджувати ботнет за допомогою свого механізму оновлення, що нагадує про те, що основне посилення безпеки має бути першим пріоритетом. Однак, оскільки DT мав автоматичний механізм оновлення, він також міг швидко виправляти пристрої, вимагаючи мінімальне втручання користувача. Впровадження автоматичних оновлень на пристроях IoT не є неможливим, але він

					ВКБР-125.26.0028.00.00.ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		
						48

повинен розроблятися з більшим усвідомленням проблематики та з більшими зусиллями інженерів.

Сповіщення. Сповіщення через позаполосні канали слугують механізмом резервного копіювання, щоб повернути мережеві пристрої у відповідність вимогам безпеки або видаляти інфекції. Приклади останніх років включають в себе сповіщення адміністраторів пристроїв через посилання CERT, зловживаючи контактом через електронну пошту у записах WHOIS, а також в попередженнях браузерів власникам сайтів про те, що їх сторінка скомпрометована. Сповіщення в просторі IoT з технічної точки зору носять складний характер. Мережевим пристроям не вистачає як публічної ознаки власності, так і встановленого каналу зв'язку для досягнення споживачів. Якщо споживачі доступні, тоді має бути чіткий і простий шлях оновлення – для вирішення проблеми. Як мінімальна альтернатива - необхідно зареєструвати адресу електронної пошти з виробником або з уніфікованою та сумісною моніторинговою платформою, яка може попереджати споживачів про серйозні проблеми. Це простір, де вимагається нетехнічне втручання. Проблема простоти використання, яка полягає в тому, щоб діяти на сповіщеннях, залишається відкритою проблемою дослідження.

Сприяння ідентифікації мережевому пристрою. Щоб покращити метод ідентифікації – виробники можуть прийняти єдиний спосіб ідентифікації версії моделі та прошивки до мережі, скажімо, кодуючи їх у частині MAC-адреси пристрою. Розкриття цієї інформації на рівні 2 зробить його видимим операторам локальної мережі (або домашньому маршрутизатору користувача), яка зможе робити автоматичні дії, щоб відключити віддалений доступ до відомого вразливого апаратного забезпечення до його оновлення. Досягнення цього в єдиній формі у всій галузі, швидше за все, вимагатиме прийняття стандартів.

Дефрагментація. Фрагментація створює ризик безпеки (і сумісності) для підтримки та управління пристроями. Під час сканування було виявлено численні реалізації Telnet, FTP та HTTP стеків. Співтовариство IoT відреагувало на це завдання, прийнявши декілька операційних систем, приклади яких

включають Android Thing, RIOT OS, Tock і Windows для IoT [30]. Цей поштовх у бік дефрагментації призведе до абстрагування від нюансів безпеки, які вимагаються від запропонованих жорстких рішень.

Термін актуальності. Навіть з урахуванням найкращих практичних методів роботи, термін служби може залишити сотні тисяч вживаних пристроїв IoT без підтримки. Відсутність довготривалої підтримки призведе до створення двокласової системи захищених та незахищених пристроїв, подібних до поточного стану машин Windows. З часом, ризик того, що ці пристрої створять більшість в Інтернеті, будуть зростати. Зупинити цей процес можна буде лише якщо такі пристрої будуть виведені в офлайн-режим.

Основною причиною компрометації та заражень пристроїв IoT є механізми оновлення. А так як вимкнення механізму оновлень або перекриття доступу пристроям IoT до мережі інтернет неможливе, через те, що це знівелює основну цінність технології та поставить їх під удар, тоді залишається лише один спосіб протидії та боротьби з ботнет у системах IoT – це аналіз мережевого трафіку, визначення загроз та блокування такого трафіку.

Для забезпечення безпеки в MQTT протоколі реалізовано такі методи захисту:

- Аутентифікація клієнтів. Пакет CONNECT може містити в собі поля USERNAME та PASSWORD. При реалізації брокера можна використовувати ці поля для автентифікації клієнта

- Контроль доступу клієнтів через Client ID

- Підключення до брокера через TLS/SSL

Спочатку ми імпортуємо бібліотеки WiFi і PubSubClient . Бібліотека WiFi може підключати ESP32 до мереж Wi-Fi, а бібліотека PubSubClient може підключати ESP32 до сервера MQTT для публікації повідомлень та підписки на теми:

```
#include <WiFi.h>
#include <PubSubClient.h>
```

					ВКБР-125.26.0028.00.00.ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		50

Встановити ім'я та пароль Wi-Fi, а також адресу з'єднання з сервером MQTT і порт і встановити тему «esp32/test»:

```
const char *ssid = "MyHomeSystem";
const char *password = "VeryCoolPass";

const char *mqtt_broker = "192.168.1.100";
const char *topic = "esp32/test";
const char *mqtt_username = "user";
const char *mqtt_password = "public";
const int mqtt_port = 1883;
```

Відкрити послідовне підключення, щоб вивести результати роботи програми:

```
Serial.begin(115200);

Підключитися до мережі Wi-Fi:
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi..");
}
```

Використати PubSubClient для підключення до брокера MQTT:

```
client.setServer(mqtt_broker, mqtt_port);
client.setCallback(callback);
while (!client.connected()) {
    String client_id = "esp32-client-";
    client_id += String(WiFi.macAddress());
    Serial.printf("The client %s connects to the mqtt broker\n", client_id.c_str());
    if (client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
        Serial.println("emqx mqtt broker connected");
    } else {
        Serial.print("Failed with state ");
        Serial.print(client.state());
        delay(2000);
    }
}
```

Після успішного підключення до сервера MQTT ESP32 опублікує повідомлення на сервері MQTT esp/test і підпишеться на повідомлення теми esp/test.

```
client.publish(topic, "Hi from EMQX-ESP32!");
client.subscribe(topic);
```

Налаштувати функцію зворотного виклику, щоб надрукувати назву теми на послідовний порт і надрукувати повідомлення, отримане з esp32/testтеми.

```
void callback(char *topic, byte *payload, unsigned int length) {  
    Serial.print("Message arrived in topic: ");  
    Serial.println(topic);  
    Serial.print("Message:");  
    for (int i = 0; i < length; i++) {  
        Serial.print((char) payload[i]);  
    }  
    Serial.println();  
    Serial.println("-----");  
}
```

Після тестування працездатності системи, потрібно зареєструвати безкоштовний профіль на MQTT сервері і додати віртуальні індикатори та пристрої до своєї системи кібербезпеки комп'ютерної мережі розумного будинку.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SEED – у криптографії симетричний блоковий криптоалгоритм на основі Мережі Фейстеля, розроблений Корейським агентством інформаційної безпеки (Korean Information Security Agency, KISA) в 1998 році. В алгоритмі використовується 128-бітний блок і ключ довжиною 128 біт. Алгоритм одержав широке поширення й використовується фінансовими й банківськими структурами, виробничими підприємствами й бюджетними установами Південної Кореї, оскільки 40-бітний SSL не забезпечує на даний момент мінімально необхідного рівня безпеки. Агентством по захисту інформації специфіковане використання шифру SEED у протоколах TLS і S/MIME. У той же час, алгоритм SEED не реалізований у більшості сучасних браузерів і інтернет-додатків, що утрудняє його використання в даній сфері поза межами Південної Кореї.

					ВКБР-125.26.0028.00.00.ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		52

SEED являє собою Мережа Фейстеля з 16 раундами, 128-бітовими блоками й 128-бітовим ключем. Алгоритм використовує дві 8×8 таблиці підстановки, які, як такі з Safer, виведені з дискретного зведення в ступінь (у цьому випадку, x^{247} і x^{251} – плюс деякі «несумісні операції»). Це є деякою подібністю с MISTY1 у рекурсивності його структури: 128-бітовий повний шифр – мережа Фейстеля з F-функцією, що впливає на 64-бітові половини, у той час як сама F-функція – Мережа Фейстеля, складена з G-функції, що впливає на 32-розрядні половини. Однак рекурсія не простягнеться далі, тому що G-функція – не Мережа Фейстеля. В G-функції 32-розрядне слово розглядають як чотири 8-бітових байта, кожний з яких проходить через одну або іншу таблицю підстановки, потім поєднується в помірковано комплексному наборі булевих функцій таким чином, що кожний біт виводу залежить від 3 з 4 вхідних байтів.

SEED має складний ключовий розклад, генеруючи тридцять два 32-розрядних додаткових символу, використовуючи G-функції на серіях обертань вихідного неопрацьованого ключа, комбінованого зі спеціальними раундовими константами (як в TEA) від «Золотого співвідношення» (англ. Golden ratio).

Згідно з дослідженнями KISA, алгоритм SEED «надійно протистоїть відомим атакам».

5. МЕТОДИКА ВПРОВАДЖЕННЯ У ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Метою розробленого програмного забезпечення є створення системи кібербезпеки мережі розумного будинку. Будь-який пристрій, підключений до мережі, може надсилати як нормальний трафік, так і трафік атаки протягом одного періоду часу. Кожен пристрій також здатний проводити різні різні атаки послідовно, а послідовні атаки можуть змінюватись за тривалістю

Алгоритм виявлення аномалій виглядає наступним чином:

1. Збір трафіку. Процес захоплення трафіку записує вихідну IP-адресу, порт джерела, IP-адресу призначення, порт призначення, розмір пакета та часовий показник всіх IP пакетів, відправлених з IoT пристроїв.

2. Групування пакетів по Пристрою та Часу. Кожен пристрій мережі розпізнається по власній IP-адресі. Пакети з кожного пристрою далі поділяються на непересічні часові проміжки, записаними межевому пристрої.

3. Витяг параметрів. Stateless та Stateful параметри створюються для кожного пакета, походження якого відоме з поведінки пристрою IoT. Stateless параметри переважно є заголовками пакетів, тоді як stateful параметри - сукупна інформація про потоки за дуже короткі часові проміжки.

Була створена експериментальна мережа пристроїв, щоб збирати типовий та зловмисний трафік. Був налаштований Raspberry Pi v3 як точка доступу Wi-Fi, щоб діяти як посередник. Потім підключена камера YI та Belkin WeMo Smart Switch до мережі Wi-Fi Raspberry Pi. Монітор артеріального тиску Myings також був підключений Bluetooth до смартфона на ОС Android, пов'язаного з мережею Wi-Fi.

Щоб зібрати звичайний (не-DoS) трафік, відбувалася взаємодія з усіма пристроями мережі протягом 10 хвилин і записувалися pcap-файли, реєструючи всі пакети, відправлені протягом цього періоду часу. Було проведено безліч

взаємодій, які трапляються під час звичайного використання пристрою, включаючи потокове відтворення відео з камери YI на сервер у режимах HD та RD, вмикання / вимикання WeMo Smart Switch та встановлення оновлень прошивки, збирання вимірювань артеріального тиску з кров'яного тиску та відправка вимірювань на сервер для зберігання. Потім було відфільтровано весь трафік, не пов'язаний з IoT, із записів pcap, у тому числі фонового трафіку з телефону Android.

Щоб уникнути ризиків безпеки та складності керування справжнім кодом ботнет, було імітувано три найпоширеніші класи DoS-атак, які запускать Mirai-інфікований пристрій: флуд протоколу TCP, SYN, UDP та HTTP GET flood. Була використана віртуальна машина Kali Linux на ноутбучі, в якості джерела DoS, а Raspberry Pi 2 запускав Apache Web Server як жертву DoS. Було з'єднано обидва пристрої за допомогою Wi-Fi, де точкою доступу виступав Raspberry Pi 3. Тоді джерело DoS націлило атаку на IP-адресу жертви. Було проведено кожен клас DoS-атак приблизно по 1,5 хвилини кожної. Точка доступу фіксувала PCAP файли трафіку атаки, за допомогою інструмента Dump-cap Linux. HTTP GET Flood був змодельований за допомогою інструмента Goldeneye. TCP SYN Flood та UDP Flood були змодельовані за допомогою утиліти hping3 Kali Linux.

Потім трафік DoS був об'єднаний із звичайним трафіком, підробляючи IP-адреси джерела, MAC-адреси та час відправки пакетів, щоб зробити його таким, як ніби пристрої IoT одночасно виробляли звичайний трафік і проводили атаки DoS. Протягом 10 хвилин кожен з трьох IoT-пристроїв, виконав кожен з трьох класів атаки DoS один раз. Атаки виконувались у випадковому порядку та випадкової тривалості, стабільно коливаючись від 90 до 110 секунд кожна. Таким чином, було зібрано приблизно 300 секунд (5 хвилин) трафіку атаки на пристрій. Розподіл атак між пристроями був незалежним.

У результаті цього процесу склав набір даних із 491 855 пакетів, що складаються з 459 655 зловмисних пакетів і 32 290 звичайних.

Досліджувалось два класи параметрів і проводився аналіз чи мають вони відношення до диференціації нормального трафіку та трафіку атаки на мережу.

					ВКБР-125.26.0028.00.00.ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		55

Stateless параметри можуть бути отримані з незалежних від потоку характеристик з окремих пакетів. Ці параметри створюються без розгрупування вхідного потоку трафіку по IP-адресі. Таким чином, ці особливості є найбільш легкими. Stateless параметри:

1. Розмір пакету. Розподіл розмірів пакетів значно відрізняється від атаки та звичайного трафіку. Більше 90% пакетів атаки становлять менше 100 байт, а звичайні - від 100 до 1200 байт. Пристрій, що проводить атаку DoS, такий як TCP SYN Flood, намагається відкрити якомога більше запитів на з'єднання з жертвою, щоб вичерпати ресурси сервера жертви. Таким чином, зловмисник хоче зберегти розмір пакетів якомога менше, щоб максимально збільшити кількість запитів на підключення в секунду. Для порівняння, звичайний трафік може варіюватися від простого пінгування серверів, які вказують на те, що пристрій активний (невеликі пакети) до даних потокового відео (великі пакети).

2. Міжпакетний інтервал. Нормальний трафік IoT має обмежене значення скупчення. Більшість пакетів надсилаються регулярними інтервалами з помітним часом між пакетами. Це може відображати мережеві зв'язки IoT або інші автоматичні дії мережі. На відміну від цього, переважна більшість трафіку атаки DoS має близькі до нуля міжпакетні інтервали (ΔT) і високі перші та другі похідні міжпакетних інтервалів.

3. Протокол. Звичайний та трафік атаки також мають різні розподіли протоколів. Трафік атаки включає менше протоколів ніж звичайний трафік та містить набір, нерелевантних до процесу роботи пристрою, протоколів.

Stateful параметри показують, як мережевий трафік розвивається з часом. У процесі генерації цих параметрів є накладні витрати, оскільки необхідно розбити мережевий трафік на потоки для кожного пристрою та розкладаємо потоки кожного пристрою у кожен проміжок часу. Stateful параметри:

1. Пропускна здатність. Алгоритм розгрупує мережевий трафік за джерелом пристрою і обчислює середню пропускну здатність протягом 10 секундних часових проміжках, щоб виміряти середню пропускну здатність

кожного пристрою. Існують незначні відмінності у пропускній здатності між звичайним та атакованим трафіком. Передбачається, що модель машинного навчання (ML) зможе використати ці відмінності.

2. IP-адреса призначення. IoT пристрої характеризуються обмеженою кількістю кінцевих точок, з якими вони спілкуються. Наприклад, комутатор WeMo спілкується лише з чотирма кінцевими точками для активації/деактивації з хмарного серверу та отримання оновлень ПЗ. Іншою ключовою характеристикою трафіку пристроїв мережі є те, що встановлена IP-адреса призначення рідко змінюється з часом. Порівняння статистики параметрів трафіку атаки та звичайного трафіку (рис.5.1):

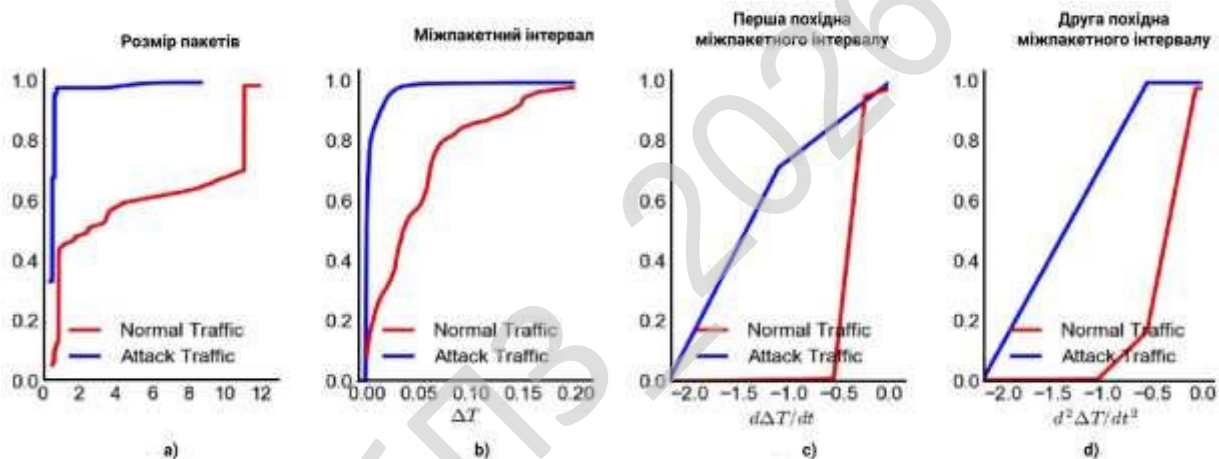


Рисунок 5.1 – Порівняння статистики параметрів трафіку атаки та звичайного трафіку. а) Розміри пакетів (b-d) Інтервали між пакетами ΔT , $d\Delta T / dt$ та $d^2\Delta T / dt^2$

Розподіл протоколів показано на рис. 5.2:

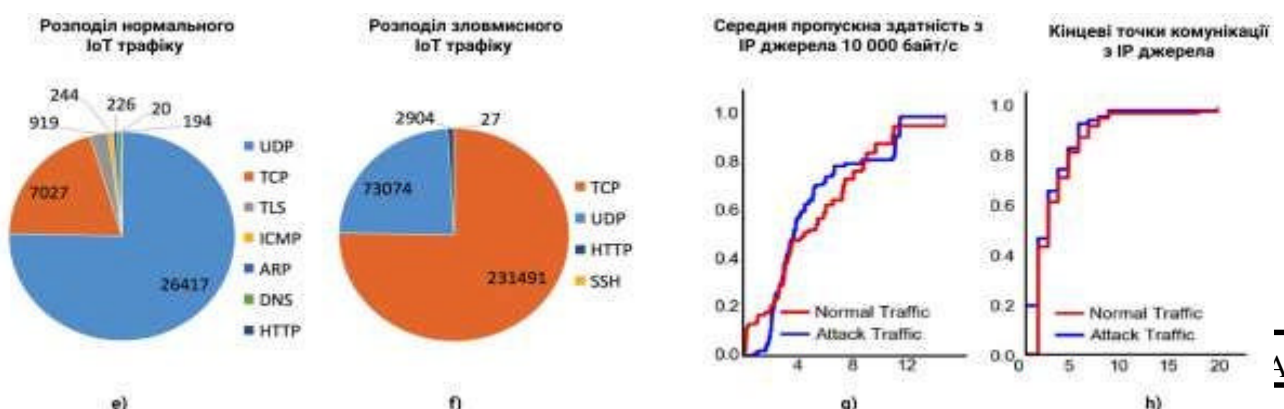


Рисунок 5.2 – Розподіл протоколів e-f – розподіл протоколів. g – середня пропускна здатність протягом більше 10 секунд. h – кількість унікальних IP-адрес у проміжку 10 секунд

Для взаємодії між собою пристрої використовують різні промислові протоколи, однією з популярних протоколів цієї мети є MQTT .

MQTT або Message Queue Telemetry Transport – це легкий, компактний та відкритий протокол обміну даними, створений для передачі даних на віддалених локаціях, де потрібний невеликий розмір коду та є обмеження щодо пропускної спроможності каналу. Перераховані вище переваги дозволяють застосовувати його в системах M2M (Машинно-Машинна взаємодія) і IoT рис.5.3:

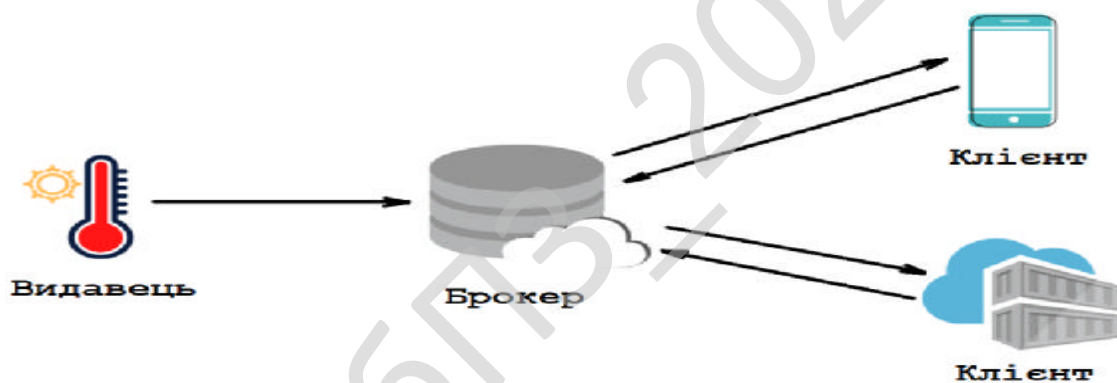


Рисунок 5.3 - Схема взаємодії за MQTT

Також існує версія протоколу MQTT-SN (MQTT for Sensor Networks), раніше відома як MQTT-S, яка призначена для бездротових пристроїв без підтримки TCP/IP мереж, наприклад, Zigbee.

Особливості протоколу MQTT

Основні особливості протоколу MQTT:

1. Асинхронний протокол
2. Компактні повідомлення
3. Робота в умовах нестабільного зв'язку на лінії передачі даних

4. Підтримка кількох рівнів якості обслуговування (QoS)
5. Легка інтеграція нових пристроїв

Ієрархія протоколів зв'язку показана на рис. 5.4:

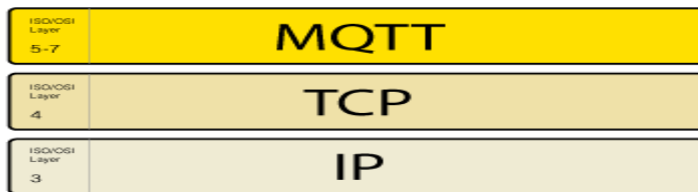


Рисунок 5.4 - Ієрархія протоколів зв'язку

Обмін повідомленнями в протоколі MQTT здійснюється між клієнтом (client), який може бути видавцем або приймачем (publisher/subscriber) повідомлень, та брокером (broker) повідомлень (наприклад, Mosquitto MQTT).

Видавець відправляє дані на брокер MQTT, вказуючи в повідомленні певну тему, топик (topic). Передплатники можуть отримувати різні дані від багатьох видавців залежно від підписки на відповідні топіки.

Пристрої MQTT використовують певні типи повідомлень для взаємодії з брокером, наведені нижче основні:

1. Connect – встановити з'єднання з брокером
2. Disconnect – розірвати з'єднання з брокером
3. Publish – опублікувати дані в топик на брокері
4. Subscribe – підписатися на топик на брокері
5. Unsubscribe – відписатися від топіка

Схема простої взаємодії (рис.5.5) між передплатником, видавцем та брокером

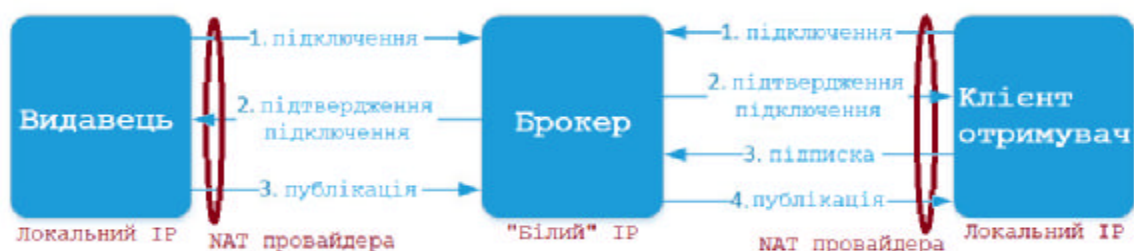


Рисунок 5.5 - Схема простої взаємодії між передплатником, видавцем та брокером

Семантика топіків

Топіки є символами з кодуванням UTF-8. Ієрархічна структура топіків має формат «дерева», що спрощує їхню організацію та доступ до даних. Топіки складаються з одного або декількох рівнів, розділених між собою символом «/».

Приклад топіка в якому датчик температури, розташований у спальній кімнаті публікує дані брокеру:

/home/living-space/living-room1/temperature

Передплатник може також отримувати дані відразу з декількох топіків, для цього існують wildcard . Вони бувають двох типів: однорівневі та багаторівневі. Для більш простого розуміння розглянемо у прикладах кожен із них. Однорівневий wildcard. Для його використання застосовується символ «+»

Багаторівневий wildcard. Для його використання застосовується символ «#».

Структура повідомлень

MQTT повідомлення складається з кількох частин:

1. Фіксований заголовок (присутнє по всіх повідомленнях)
2. Змінний заголовок (є лише у певних повідомленнях)
3. Дані, «навантаження» (є лише у певних повідомленнях)

Розглянемо (рис.5.6) фіксований заголовок:

Bit	7	6	5	4	3	2	1	0
Byte 1	Message Type				Flags specific to each MQTT packet			
Byte 2	Remaining Length							

Рисунок 5.6 - Фіксований заголовок MQTT пакету

Message Type – це тип повідомлення, наприклад: CONNECT, SUBSCRIBE, PUBLISH та інші.

Flags specific to each MQTT packet – ці 4 біти відведені під допоміжні прапори, наявність та стан яких залежить від типу повідомлення.

Remaining Length - представляє довжину поточного повідомлення (змінний заголовок + дані), може займати від 1 до 4 байти.

Загалом у протоколі MQTT існує 15 типів повідомлень.

Позначки ознак пакетів (прапори) займають чотири старші біти (рис5.7) першого байта фіксованого заголовка:

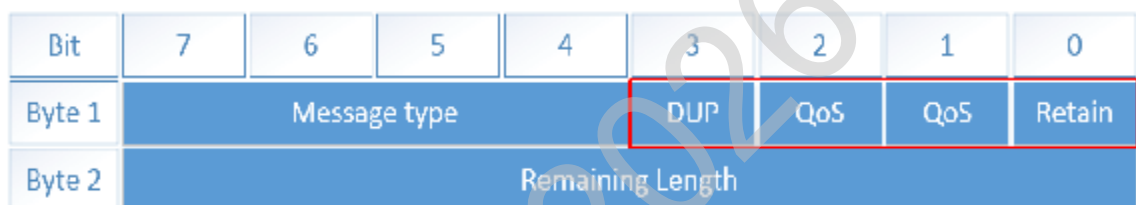


Рисунок 5.7 - Прапори в першому байті повідомлення

1. DUP – прапорець дублікату встановлюється, коли клієнт або MQTT брокер здійснює повторне відправлення пакета (використовується у типах PUBLISH, SUBSCRIBE, UNSUBSCRIBE, PUBREL). Якщо встановлено прапорець, змінний заголовок повинен містити Message ID (ідентифікатор повідомлення).

2. QoS – якість обслуговування (0,1,2).

3. RETAIN – при публікації даних із встановленим прапором retain, брокер збереже його. При наступній підписці на цей топик брокер негайно надішле повідомлення із цим прапором. Використовується лише у повідомленнях із типом PUBLISH.

Змінний заголовок (рис. 5.7) міститься в деяких заголовках. У ньому містяться такі дані:

1. Packet identifier – ідентифікатор пакета, присутній у всіх типах повідомлень, крім: CONNECT, CONNACK, PUBLISH (з QoS <1), PINGREQ, PINGRESP, DISCONNECT

2. Protocol name – назва протоколу (тільки у повідомленнях типу CONNECT)

3. Protocol version – версія протоколу (тільки у повідомленнях типу CONNECT)

4. Connect flags – прапори, що вказують на поведінку клієнта при підключенні

5.

Bit	7	6	5	4	3	2	1	0
Byte 8	User name	Password	Will Retain	Will QoS		Will Flag	Clean Session	Reserved

Рисунок 5.7 - Змінний заголовок

1. User name – за наявності цього прапора у «навантаженні» має бути вказано ім'я користувача (використовується для автентифікації клієнта)

2. Password – за наявності цього прапора у «навантаженні» має бути вказаний пароль (використовується для автентифікації клієнта)

3. Will Retain - при установці в 1, брокер зберігає у себе Will Message.

4. Will QoS – якість обслуговування для Will Message, при встановленому прапорі Will Flag, Will QoS та Will retain є обов'язковими.

5. Will Flag - при встановленому прапорі, після того, як клієнт відключиться від брокера без надсилання команди DISCONNECT (у випадках непередбачуваного обриву зв'язку тощо), брокер сповістить про це всіх підключених клієнтів через так званий Will Message.

6. Clean Session – очистити сесію. При встановленому "0" брокер збереже сесію, всі підписки клієнта, а також передасть йому всі повідомлення з QoS1 і QoS2, які були отримані брокером під час відключення клієнта, при наступному

підключенні. Відповідно при встановленій «1», при повторному підключенні клієнту необхідно буде заново підписуватися на топіки.

Session Present – використовується у повідомленні з типом CONNACK. Якщо брокер приймає підключення з Clean Session = 1, він повинен встановити «0» в біт Session Present(SP). Якщо брокер приймає підключення з Clean Session = 0, то значення біта SP залежить від того, чи брокер зберігав раніше сесію з цим клієнтом (якщо так, то в SP виставляється 1 і навпаки). Тобто цей параметр дозволяє клієнту визначити, чи була збережена брокером попередня сесія.

Connect Return code – якщо брокер з якихось причин не може прийняти правильно сформований CONNECT пакет від клієнта, то в другому байті CONNACK пакета він повинен встановити відповідне значення з переліку нижче (рис. 5.8):

Значення	Повернене значення	Опис
0	0x00 Connection Accepted	Підключення прийнято
1	0x01 Connection Refused, unacceptable protocol version	Брокер не підтримує версію протоколу, що використовується клієнтом
2	0x02 Connection Refused, identifier rejected	Client ID клієнта, що підключається, немає в списку дозволених
3	0x03 Connection Refused, Server unavailable	З'єднання встановлено, але MQTT сервіс недоступний
4	0x04 Connection Refused, bad user name or password	Не правильний логін чи пароль
5	0x05 Connection Refused, no authorized	Доступ до підключення заборонено
6-255		зарезервовано

Рисунок 5.8 - Коды помилок, які повертає брокер

MQTT підтримує три рівні якості обслуговування (QoS) під час надсилання повідомлень. QoS 0 At most once . На цьому рівні видавець один раз відправляє повідомлення брокеру (рис.5.9) і не чекає на підтвердження від нього, тобто відправив і забув.



Рисунок 5.9 - Взаємодія без забезпечення якості обслуговування (QoS)

QoS 1 At least once . Цей рівень гарантує, що повідомлення точно буде доставлено брокеру, але є можливість дублювання повідомлень від видавця. Після отримання дубліката повідомлення брокер знову розсилає це повідомлення передплатникам, а видавцеві знову надсилає підтвердження про отримання повідомлення. Якщо видавець не отримав повідомлення PUBACK від брокера, він повторно відправляє цей пакет, при цьому в DUP встановлюється «1».

QoS 2 Exactly once. На цьому рівні гарантується надсилання повідомлень клієнту та виключається можливе дублювання надісланих повідомлень.

Видавець надсилає повідомлення брокеру. У цьому повідомленні вказується унікальний пакет ID, QoS=2 і DUP=0. Видавець зберігає повідомлення непідтвердженим, поки не отримає від брокера відповідь PUBREC. Брокер відповідає повідомленням PUBREC, в якому міститься той же Packet ID. Після отримання видавець відправляє PUBREL з тим же Packet ID. До того, як брокер отримає PUBREL, він повинен зберігати копію повідомлення в себе. Після отримання PUBREL він видаляє копію повідомлення та надсилає видавцеві повідомлення PUBCOMP про те, що транзакція завершена.

Для забезпечення безпеки в MQTT протоколі реалізовано такі методи захисту:

- Аутентифікація клієнтів. Пакет CONNECT може містити в собі поля USERNAME та PASSWORD. При реалізації брокера можна використовувати ці поля для автентифікації клієнта

- Контроль доступу клієнтів через Client ID
- Підключення до брокера через TLS/SSL

Спочатку ми імпортуємо бібліотеки WiFi і PubSubClient . Бібліотека WiFi може підключати ESP32 до мереж Wi-Fi, а бібліотека PubSubClient може підключати ESP32 до сервера MQTT для публікації повідомлень та підписки на теми:

```
#include <WiFi.h>
#include <PubSubClient.h>
```

Встановіть ім'я та пароль Wi-Fi, а також адресу з'єднання з сервером MQTT і порт і встановіть тему «esp32/test»:

```
const char *ssid = "MyHomeSystem";
const char *password = "VeryCoolPass";

const char *mqtt_broker = "192.168.1.100";
const char *topic = "esp32/test";
const char *mqtt_username = "user";
const char *mqtt_password = "public";
const int mqtt_port = 1883;
```

Відкрити послідовне підключення, щоб вивести результати роботи програми:

```
Serial.begin(115200);
```

```
Підключитися до мережі Wi-Fi:
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi..");
}
```

Використати PubSubClient для підключення до брокера MQTT:

```
client.setServer(mqtt_broker, mqtt_port);
client.setCallback(callback);
while (!client.connected()) {
    String client_id = "esp32-client-";
    client_id += String(WiFi.macAddress());
    Serial.printf("The client %s connects to the mqtt broker\n",
client_id.c_str());
    if (client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
        Serial.println("emqx mqtt broker connected");
    } else {
        Serial.print("Failed with state ");
        Serial.print(client.state());
        delay(2000);
    }
}
```

Після успішного підключення до сервера MQTT ESP32 опублікує повідомлення на сервері MQTT esp/test і підпишеться на повідомлення теми esp/test.

```
client.publish(topic, "Hi from EMQX-ESP32!");
client.subscribe(topic);
```

Налаштуйте функцію зворотного виклику, щоб надрукувати назву теми на послідовний порт і надрукувати повідомлення, отримане з esp32/test теми.

```
void callback(char *topic, byte *payload, unsigned int length) {
    Serial.print("Message arrived in topic: ");
    Serial.println(topic);
    Serial.print("Message:");
```

```

for (int i = 0; i < length; i++) {
    Serial.print((char) payload[i]);
}
Serial.println();
Serial.println("-----");
}

```

Після тестування працездатності системи, потрібно зареєструвати безкоштовний профіль на MQTT сервері і додати віртуальні індикатори та пристрої до своєї системи кібербезпеки мережі розумного будинку.

КБПЗ_2026

					ВКБР-125.26.0028.00.00.ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		66

6 ОСНОВНІ ВИСНОВКИ

У даній кваліфікаційній бакалаврській роботі створено програмне забезпечення системи кібербезпеки комп'ютерної мережі розумного будинку.

Проаналізовано, які задачі виконує система кібербезпеки і навіщо її застосовувати, а також висвітлено головні проблеми системи кібербезпеки комп'ютерної мережі розумного будинку. У підсумку це дозволило обрати необхідні протоколи та технології, на яких буде базуватися майбутня система.

У роботі запропоновано систему кібербезпеки комп'ютерної мережі розумного будинку на основі мікроконтролера Raspberry Pi.

Система може контролювати стан комп'ютерної мережі через інтернет, коли оновлення інформації на веб-сервері зчитується за розробленим алгоритмом, що подається в Raspberry Pi, а потім система відповідає на відповідні інструкції з високим ступенем захисту. Користувач може безпосередньо входити в систему та взаємодіяти із пристроями у режимі реального часу.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення.

Система є енергоефективною, інтелектуальною, має низьку вартість, портативність та високу продуктивність.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Довгий С.О., Савченко О.Я., Воробієнко П.П. та ін. Сучасні телекомунікації: мережі, технології, економіка, управління, регулювання / За ред. С.О. Довгого. – К.: Український Видатничий Центр, 2002. – 520 с.
2. Телекомунікаційні та інформаційні мережі : Підручник [для вищих навчальних закладів] / П.П. Воробієнко, Л.А. Нікітюк, П.І. Резніченко. – К.: САММІТ-Книга, 2010. – 708 с.: іл.
3. CCNA R&S RSE Chapter 11: Network Address Translation for IPv4 // Електронний ресурс. Режим доступу: <https://static-courseassets.s3.amazonaws.com/RSE50ENU/module11/index.html>
4. Грайворонський М. В. Безпека інформаційно-комунікаційних систем / М. В. Грайворонський, О. М. Новіков – К.: Видавнича група BHV, 2009. – 608 с.
5. NIST SP 800-22rev1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications//National Institute of Standards and Technology Special Publication 800-22rev1a, 2010, - 131 p.
6. Комп'ютерні мережі [навчальний посібник] / А.І.Блозва, Ю.В.Матус, В.В.Смолій, Б.С.Гусев, Д.Ю.Касаткін, Т.Ю.Осипова, Я.А.Савицька // - К.: Компрінт, 2017.- 821с.
7. ABRAMSON, N.: "Internet Access Using VSATs," IEEE Commun. Magazine, vol. 38, pp. 60–68, July 2000.
8. E. Eskin, W. Lee, and W. Stolfo – «Modeling system call for intrusion detection using dynamic window sizes»
9. M. Qin and K. Hwang – «Frequent episode rules for internet anomaly detection»
10. Фіногенов К.Г. Системні функції MS-DOS – К.: 1997. –390с.
11. Нортон П., Уілтон Р. IBM PC та PS/2. Посібник по програмуванню: Пер. з англ. – К.: Радіо і зв'язок, 1994. – 336 с.
12. Борн Г. Формати даних: Пер. з нім. – К.: BHV, 1995 – 472 с.

13. Федоров А., Рогаткін Дм. Borland Pascal в середовищі Windows – Київ: “Діалектика”, 1993. – 656 с.
14. Фрід Е., Пастор І., Ревес П., Рейман Й., Ружа І. Мала математична енциклопедія – Пер. з венг.: 1976, - 693 с.
15. Hewlett Packard Enterprise – «Internet of things research study». URL: <http://h20195.www2.hp.com/V4/getpdf.aspx/4aa5-4759enw>
16. «Internet of things (iot) security and privacy recommendations.».
17. S. Hilton – «Dyn analysis summary of friday october 21 attack.» URL: <https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/>
18. V.Chandola, A.Banerjee, V.Kumar – «Anomaly detection: A survey» vol. 41.3
19. Довгий С.О., Савченко О.Я., Воробієнко П.П. та ін. Сучасні телекомунікації: мережі, технології, економіка, управління, регулювання / За ред. С.О. Довгого. – К.: Український Видатничий Центр, 2002. – 520 с.
20. Телекомунікаційні та інформаційні мережі : Підручник [для вищих навчальних закладів] / П.П. Воробієнко, Л.А. Нікітюк, П.І. Резніченко. – К.: САММІТ-Книга, 2010. – 708 с.: іл.
21. CCNA R&S RSE Chapter 11: Network Address Translation for IPv4 // Електронний ресурс. Режим доступу: <https://static-courseassets.s3.amazonaws.com/RSE50ENU/module11/index.html>
22. Грайворонський М. В. Безпека інформаційно-комунікаційних систем / М. В. Грайворонський, О. М. Новіков – К.: Видавнича група BVH, 2009. – 608 с.
23. NIST SP 800-22rev1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications//National Institute of Standards and Technology Special Publication 800-22rev1a, 2010, - 131 p.
24. Комп’ютерні мережі [навчальний посібник] / А.І.Блозва, Ю.В.Матус, В.В.Смолій, Б.С.Гусєв, Д.Ю.Касаткін, Т.Ю.Осипова, Я.А.Савицька // - К.: Компрінт, 2017.- 821с.
25. ABRAMSON, N.: “Internet Access Using VSATs,” IEEE Commun. Magazine, vol. 38, pp. 60–68, July 2000.

26. AHMADI, S.: “An Overview of Next-Generation Mobile WiMAX Technology,” IEEE Commun.Magazine, vol. 47, pp. 84–88, June 2009.
27. ALLMAN, M., and PAXSON, V.: “On Estimating End-to-End Network Path Properties,” Proc.SIGCOMM '99 Conf., ACM, pp. 263–274, 1999.
28. ANDERSON, C.: The Long Tail: Why the Future of Business is Selling Less of More, rev. upd. ed.,New York: Hyperion, 2008a.
29. ANDERSON, R.J.: Security Engineering: A Guide to Building Dependable Distributed Systems, 2nded., New York: John Wiley & Sons, 2008b.
30. ANDERSON, R.J.: “Free Speech Online and Office,” Computer, vol. 25, pp. 28–30, June 2002.
31. ANDERSON, R.J.: “The Eternity Service,” Proc. Pragocrypt Conf., CTU Publishing House, pp.242–252, 1996.
32. ANDREWS, J., GHOSH, A., and MUHAMED, R.: Fundamentals of WiMAX: Understanding Broadband Wireless Networking, Upper Saddle River, NJ: Pearson Education, 2007.
33. ASTELY, D., DAHLMAN, E., FURUSKAR, A., JADING, Y., LINDSTROM, M., and PARKVALL, S.: “LTE: The Evolution of Mobile Broadband,” IEEE Commun. Magazine, vol. 47, pp. 44–51, Apr. 2009.
34. BALLARDIE, T., FRANCIS, P., and CROWCROFT, J.: “Core Based Trees (CBT),” Proc. SIGCOMM' 93 Conf., ACM, pp. 85-95, 1993.
35. BARAN, P.: “On Distributed Communications: I. Introduction to Distributed Communication Networks,” Memorandum RM-420-PR, Rand Corporation, Aug. 1964.
36. BELLAMY, J.: Digital Telephony, 3rd ed., New Ytfrk: Wiley, 2000.
37. BELLMAN, R.E.: Dynamic Programming, Princeton, NJ: Princeton University Press, 1957.
38. BELLOVIN, S.: “The Security Flag in the IPv4 Header,” RFC 3514, Apr. 2003.
39. BELSNES, D.: “Flow Control in the Packet Switching Networks,” Communications Networks, Uxbridge, England: Online, pp. 349–361, 1975.

40. BENNET, C.H., and BRASSARD, G.: “Quantum Cryptography: Public Key Distribution and Coin Tossing,” Int’l Conf. on Computer Systems and Signal Processing, pp. 175–179, 1984.

41. BERESFORD, A., and STAJANO, F.: “Location Privacy in Pervasive Computing,” IEEE Pervasive Computing, vol. 2, pp. 46–55, Jan. 2003.

42. BERGHEL, H.L.: “Cyber Privacy in the New Millennium,” Computer, vol. 34, pp. 132–134, Jan. 2001.

43. BERNERS-LEE, T., CAILLIAU, A., LOUTONEN, A., NIELSEN, H.F., and SECRET, A.: “The World Wide Web,” Commun. of the ACM, vol. 37, pp. 76– 82, Aug. 1994.

44. BERTSEKAS, D., and GALLAGER, R.: Data Networks, 2nd ed., Englewood Cliffs, NJ: Prentice Hall, 1992.

45. BHATTI, S.N., and CROWCROFT, J.: “QoS Sensitive Flows: Issues in IP Packet Handling,” IEEE Internet Computing, vol. 4, pp. 48–57, July–Aug. 2000.

46. BIHAM, E., and SHAMIR, A.: “Differential Cryptanalysis of the Data Encryption Standard,” Proc. 17th Ann. Int’l Cryptology Conf., Berlin: SpringerVerlag LNCS 1294, pp. 513–525, 1997.

47. BIRD, R., GOPAL, I., HERZBERG, A., JANSON, P.A., KUTTEN, S., MOLVA, R., and YUNG, M.: “Systematic Design of a Family of AttackResistant Authentication Protocols,” IEEE J. on Selected Areas in Commun., vol. 11, pp. 679–693, June 1993.

48. BIRRELL, A.D., and NELSON, B.J.: “Implementing Remote Procedure Calls,” ACM Trans, on Computer Systems, vol. 2, pp. 39–59, Feb. 1984.

49. BIRYUKOV, A., SHAMIR, A., and WAGNER, D.: “Real Time Cryptanalysis of A5/1 on a PC,” Proc. Seventh Int’l Workshop on Fast Software Encryption, Berlin: Springer-Verlag LNCS 1978, p. 1, 2000.

50. BLAZE, M., and BELLOVIN, S.: “Tapping on My Network Door,” Commun. of the ACM, vol. 43, p. 136, Oct. 2000.936

51. NAT на Cisco // Электронный ресурс. Режим доступа: <https://habrahabr.ru/post/108931/>

					ВКБР-125.26.0028.00.00.ПЗ		Арк.
Зм.	Арк.	№ документа	Підпис	Дата			71

52. How Network Address Translation Works // Электронный ресурс. Режим доступа: <http://computer.howstuffworks.com/nat.htm>
53. B.Eckel Thinking in Java, 2nd ed. Prentice Hall PTR; 2nd Bk&cdr edition, 2000.
54. Bill Venners Inside the Java 2 Virtual Machine. Osborne McGraw-Hill; 2nd Bk&cdr edition, 2000.
55. Tim Lindholm, Frank Yellin The Java(TM) Virtual Machine Specification. Addison-Wesley Pub Co; 2nd edition, 1999.
56. Ed Roman, Scott W. Ambler, Tyler Jewell, Floyd Marinescu Mastering Enterprise JavaBeans. John Wiley & Sons; 2nd Bk&acc edition, 2001.
57. W. Leonhard, "Control of electrical drives", 2nd Ed, Springer, 1996.
58. F.A. Toliyat, S.G. Campbell, "DSP-based electromechanical motion control", CRC Press, 2004.
59. Botchkaryov, A., Golembo, V., Paramud, Y., Yatsyuk, V. (2019). Cyber-physical systems: data collection technologies, A. Melnyk (ed.), Lviv: Magnolia 2006. 176 p. (in Ukrainian). ISBN: 98-617-574-139-9.
60. Smart dispatcher [Electronic resource]. Available at: <https://hytera-europe.com/news/what-is-smart-dispatch>, (Accessed: 29/09/2023).
61. Power system analysis [Electronic resource]. Available at: https://cdn.regulation.gov.ua/regulation.gov.ua_GP%20Electricity%20market.pdf (Accessed: 29/09/2023).
62. Paramud Y., Butov M., Pavych T. (2021). Features of the implementation of smart technologies in the mine // Advances in Cyber-Physical Systems, Vol. 6, No 2, 98–104. DOI: <https://doi.org/10.23939/acps2021.02.098>.
63. Vulnerabilities in smart home systems [Electronic resource]. Available at: <https://cybercalm.org/novyny/urazlyvosti-v-systemah-rozumnogo-domu-dozvlyayut-hakeram-vas-pidsluhovuvaty/>, (Accessed: 29/09/2023).
64. Imran A., Atif S. (2018). Comprehensive Energy Systems, Vol. 5, 389–422. DOI: <https://doi.org/10.1016/B978-0-12-809597-3.00524-1>

65. Nataliia Pavych, Tetyana Pavych, “Method for Time Minimisation of API Requests Service From Cyber-Physical System to Cloud Database Management System“. Advances in Cyber-Physical Systems, 2019, Vol. 4, No. 2, 125–131. DOI: <https://doi.org/10.23939/acps2019.02.125>.

66. ZigBee protocol [Electronic resource] – Available at: <https://www.geeksforgeeks.org/introduction-of-zigbee/>, (Accessed: 30/01/2023).

67. Wi-Fi: IEEE 802.11 [Electronic resource]. Available at: <https://www.electronics-notes.com/articles/connectivity/wifi-ieee-802-11/what-is-wifi.php>, (Accessed: 29/09/2023).

68. Bluetooth [Electronic resource]. Available at: <https://www.britannica.com/technology/Bluetooth>, (Accessed: 29/09/2023). 11. Data Sheet Bluetooth Low Energy Module RN4020 [Electronic resource]. Available at: <https://ww1.microchip.com/downloads/en/DeviceDoc/50002279B.pdf>, (Accessed: 29/09/2023).