Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки та захисту інформації
_____ Іван ПАРХОМЕНКО
«____» червня 2023 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)
спеціальність 125 Кібербезпека
(код і назва спеціальності)
освітній ступень бакалавр
освітня програма Кібербезпека
(назва освітньо-професійної програми)
на тему: Механізми захисту складових системи інтернету речей

Виконавець: студент IV курсу, групи КБ-41

_____ **Родіон КОДЖЕБАШ**
(підпис) (ім'я, прізвище)

	Ім'я, прізвище	**Підпис**
Керівник	Інна МИХАЛЬЧУК	

Нормоконтроль	Андрій БІГДАН	

Київ 2023

Міністерство освіти і науки України

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри кібербезпеки
та захисту інформації
_____ Сергій ТОЛЮПА
«24» жовтня 2022 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньо-професійної програми)

Студенту **КБ-41** (група) **Коджебаш Родіону Михайловичу** (прізвище ім'я по батькові)

Тема кваліфікаційної роботи Механізми захисту складових системи інтернету речей .

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №3 від 20.10.2022 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Безпека розумних будинків, протоколи зв'язку в розумних будинках, технології захисту складових систем IoT.

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Аналіз основних підсистем складових IoT і дослідження історії виникнення систем IoT. Аналіз загальнодоступної інформації про кібератаки. Наведення підходів для запобігання кібератак на системи IoT та зменшення збитків від інцидентів та витоків інформації.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Результати дослідження важливі для забезпечення кібербезпеки

«розумних будинків» та для покращення методів захисту «розумного будинку»

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Завдання видала _____ (підпис) Інна МИХАЛЬЧУК (ім'я, прізвище)

Завдання прийняв до виконання _____ (підпис) Родіон КОДЖЕБАШ (ім'я, прізвище)

Дата видачі завдання: 24 жовтня 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Отримання завдання	25.10.2022 – 28.11.2022	*виконано*
2	Аналіз літератури	29.11.2022 – 03.01.2023	*виконано*
3	Огляд системи розумний будинок	04.01.2023 –19.01.2023	*виконано*
4	Збір відомостей щодо системи запобігання вторгненням	22.01.2023 – 27.03.2023	*виконано*
5	Дослідження основних елементів IoT	28.03.2023 – 15.04.2023	*виконано*
6	Аналіз основних вразливостей в складових системах IoT	15.04.2023 – 02.05.2023	*виконано*
7	Написання тексту атестаційної роботи	03.05.2023 – 16.05.2023	*виконано*
8	Оформлення пояснювальної записки	16.05.2023 – 01.06.2023	*виконано*
9	Підготовка та оформлення роботи до захисту	02.06.2023 – 13.06.2023	*виконано*

Завдання видала _____ (підпис) Інна МИХАЛЬЧУК (ім'я, прізвище)

Завдання прийняв до виконання _____ (підпис) Родіон КОДЖЕБАШ (ім'я, прізвище)

Термін подання кваліфікаційної роботи до ЕК 12 червня 2023 року

РЕФЕРАТ

Кваліфікаційна робота складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел, двох додатків, має 67 сторінок основного тексту, 13 рисунків. Список використаних джерел містить 25 найменувань і займає 3 сторінки.

Об’єктом дослідження механізми захисту складових системи інтернету речей. **Предметом дослідження** є елементи IoT відповідальні за захист системи. **Методи дослідження** використанні при підготовці кваліфікаційної роботи:

• аналіз наукової літератури;

• розробка додатку безпеки;

• порівняння та синтез;

• аналіз даних.

У даній кваліфікаційній роботі було проведено комплексне дослідження з метою підвищення безпеки складових систем IoT. По-перше, був проведений аналіз наукової літератури, що стосується складових систем IoT та методів виявлення та запобігання вторгнень. Цей аналіз включав вивчення ключових методів та алгоритмів, пов'язаних з безпекою систем IoT.

Результати досліджень можуть бути застосовані в сфері інформаційної безпеки для поліпшення методів захисту складових систем IoT від кібератак. Наприклад, на основі отриманих даних можуть бути розроблені додатки для підвищення захисту від кібератак.

Практична цінність отриманих результатів полягає в розробці програмного забезпечення для захисту каналів зв’язку BlueTooth від втручання до цільової машини або пристроїв IoT та перехоплення даних.

Напрямки подальших досліджень: розробка нових методів підвищення захисту безпеки складових систем IoT від кібератак на основі отриманих результатів дослідження.

Ключові слова: IoT, методи захисту, аналіз даних, вектор атак, безпека мережі,

BlueTooth, системи контролю доступу, класифікація атак, інформаційна безпека.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

IoT – Internet of things

Wi-Fi – Wireless fidelity

BLE – BlueTooth low energy

IP – Internet protocol

ISP – Internet service provider

IT – Information Technology

IDS – Intrusion detection system

CIA – Confidentiality, integrity, availability,

PLS – Programmable Logic Controller

CSA – Connectivity Standards Alliance

WPAN – Wireless personal area network

LAN – Local Area Network

SIEM – Security Information And Event Management TCP – Transmission Control Protocol

IPv6 – Internet Protocol version 6

DNS – Системи доменних імен

ЗМІСТ

ВСТУП..... 8
РОЗДІЛ 1 ОГЛЯД СКЛАДОВИХ СИСТЕМИ ІОТ 13
1.1 Аналіз архітектури IoT 13
1.2 Вивчення методів забезпечення конфіденційності інформації..... 16
1.3 Дослідження принципів функціонування кожної складової системи..... 21
РОЗДІЛ 2 ЗАГРОЗИ БЕЗПЕЦІ СКЛАДОВИХ СИСТЕМИ ІОТ..... 25
2.1 Класифікація загроз за походженням та способами реалізації 25
2.2 Розгляд відомих прикладів атак на системи IoT..... 29
РОЗДІЛ 3 МЕХАНІЗМИ ЗАХИСТУ СКЛАДОВИХ СИСТЕМИ ІОТ 33
3.1 Методи аутентифікації та авторизації 33
3.2 Застосування шифрування для захисту даних 36
3.3 Методи контролю доступу до системи..... 40
3.4 Системи моніторингу та логування 43
3.5 Фізичний захист складових системи IoT..... 47
РОЗДІЛ 4 РОЗРОБКА ДОДАТКУ НА БАЗІ BLUETOOTH ТА ВИСНОВКИ 51
4.1 Розробка додатку, опис принципу та роботи 51
4.2 Висновки щодо ефективності застосування механізмів захисту в цілому 60
ВИСНОВКИ..... 63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... 65
ДОДАТОК А..... 68
Лістинг А.1 Фрагмент програмного коду файлу 68
Лістинг А.2 Фрагмент програмного коду файлу 71

ВСТУП

З інтернетом речей (IoT) пов'язано безліч переваг, проте водночас існують серйозні проблеми щодо безпеки і захисту персональних даних користувачів. Оскільки система IoT складається з великої кількості різноманітних компонентів, які співпрацюють між собою, зламати один компонент може призвести до порушення безпеки всієї системи. Тому необхідно вживати заходів з метою забезпечення захисту системи IoT. Механізми захисту можуть бути різноманітними, від криптографічних протоколів до мережевих фільтрів.

Важливо також враховувати можливість виникнення нових загроз і адаптувати заходи захисту до них. У цьому контексті, ця тема досліджує механізми захисту складових системи IoT та їхню роль у забезпеченні безпеки інтернету речей в цілому. Вона є дуже актуальною, оскільки високий рівень безпеки IoT є критичним для забезпечення конфіденційності і приватності користувачів, а також для захисту від зловмисних атак. Забезпечення безпеки в системах IoT є важливою проблемою в сучасному світі, де інтернет речей стає все більш поширеним. Зловмисники можуть використовувати ці системи для різних цілей, таких як збір інформації про користувачів, атаку на мережеві ресурси або керування з'єднаннями IoT пристроїв.

Механізми захисту систем IoT мають на меті забезпечити захист пристроїв від несанкціонованого доступу, захист мережевих з'єднань від атак і виявлення можливих загроз. Це включає в себе різноманітні техніки, такі як шифрування, аутентифікацію, авторизацію, захист від переповнення буфера, зміну адрес та інші. Одним з найважливіших механізмів захисту в системах IoT є криптографічне шифрування. Воно забезпечує конфіденційність даних, захищає їх від зламування та забезпечує автентифікацію та інтегритет даних. Також важливо використовувати бездротові протоколи з захистом від атак. Для забезпечення високого рівня безпеки IoT, необхідно розробляти і використовувати мережеві фільтри, які допоможуть зменшити можливість входження зловмисників у мережу. Такі фільтри можуть блокувати небажані з'єднання, відокремлювати мережеві сегменти та обмежувати доступ до

ресурсів. Крім того, важливо використовувати комплексний підхід до захисту систем IoT, який включає в себе як технічні, так і організаційні заходи. Наприклад, дотримання протоколів безпеки при розробці та експлуатації систем IoT, проведення навчання та свідомості серед користувачів про можливі загрози та заходи безпеки.

Враховуючи швидкий розвиток технологій та зростання використання IoT в різних галузях, забезпечення безпеки в системах IoT стає все більш важливим. Тому розуміння механізмів захисту в системах IoT та їх впровадження має велике значення для забезпечення безпеки та приватності користувачів. Недостатня безпека в системах IoT може мати серйозні наслідки для користувачів, включаючи можливість зламування пристроїв та витоку особистої інформації. Наприклад, зламник може використовувати відомості з IoT-пристроїв для здійснення кібератак, крадіжки особистої інформації, або зламувати системи керування важливими інфраструктурними об'єктами.

Для забезпечення безпеки в системах IoT необхідно враховувати особливості таких систем, зокрема велику кількість підключених пристроїв, їх різноманітність та широкий спектр використання. Також необхідно розглядати потенційні загрози, зокрема злами, віруси, DDoS-атаки, крадіжку ідентифікаторів користувачів та інформації. Для захисту систем IoT використовуються різні механізми, включаючи аутентифікацію та авторизацію, шифрування даних, мережеві брандмауери та системи виявлення вторгнень. Однак, найефективнішим підходом є комплексний підхід до захисту, що включає в себе усі вище перераховані заходи. Технічні заходи можуть включати в себе розробку безпечних протоколів обміну даними, захист мережі та пристроїв від атак, шифрування даних, використання біометричних технологій та інші заходи. Тому розуміння механізмів захисту в системах IoT та їх впровадження має велике значення для забезпечення безпеки та приватності користувачів.

Актуальність:

Тема механізмів захисту складових системи Інтернету речей є дуже актуальною в наш час, коли зростає кількість підключених до мережі пристроїв IoT, а також збільшується обсяг збереженої на них конфіденційної інформації. Це створює серйозні виклики в галузі кібербезпеки, оскільки системи IoT мають деякі специфічні властивості, які роблять їх більш вразливими до кібератак. Зокрема, це стосується

більшої кількості точок входу, що можуть бути атаковані, нестабільної роботи пристроїв, обмежених ресурсів пам'яті та енергопостачання, недостатньої захисту від різних типів атак. Крім того, з розвитком технологій інтернету речей, з'являються нові можливості для кіберзлочинців, такі як віддалене керування пристроями, збір та використання конфіденційної інформації, зміна налаштувань пристроїв і т.д. У зв'язку зі зростанням кількості підключених до Інтернету речей пристроїв, ризики зламу та кібератак на системи IoT зростають.

Це може становити серйозну загрозу як для окремих користувачів, так і для бізнесу та громадського сектору. Забезпечення кібербезпеки в системах IoT вимагає розробки та впровадження комплексної системи захисту, яка включає в себе як технічні, так і організаційні механізми. Технічні механізми захисту включають в себе шифрування, аутентифікацію та авторизацію, контроль доступу, моніторинг та аналіз безпекових подій та інші методи, що дозволяють забезпечити захист пристроїв та їх даних від кіберзлочинців.

Організаційні механізми захисту включають в себе політики безпеки, процедури аудиту безпеки, навчання та свідоме використання користувачами безпечних практик. Наприклад, навчання користувачів про безпеку паролів та заборону використовувати однакові паролі на кількох пристроях, може зменшити ризики зламу від зловмисників. Крім того, механізми захисту в системах Інтернету речей є важливими з точки зору захисту від кібертероризму. У сучасному світі кібертерористи можуть використовувати різні методи атак на системи IoT, такі як DDOS атаки, розповсюдження шкідливого програмного забезпечення, фішинг та соціальні інженерні атаки.

Ці атаки можуть призвести до паралізування функціонування пристроїв Інтернету речей, зниження їхньої надійності та стійкості до збоїв, а також до втрати даних та витоку конфіденційної інформації. Іншим аспектом актуальності цієї теми є швидке зростання кількості пристроїв Інтернету речей, які використовуються в різних галузях, таких як транспорт, промисловість, охорона здоров'я та інші. Ці пристрої збирають та обробляють великі обсяги даних, що може призвести до серйозних проблем з захистом конфіденційної інформації та забезпечення безпеки користувачів. Отже, розробка та впровадження ефективних механізмів захисту в системах Інтернету

речей є надзвичайно важливою задачею, що дозволить зменшити ризики зламу та кібератак, забезпечити безпеку та конфіденційність пристроїв та інформації, що на них зберігається, а також забезпечити стійкість та надійність пристроїв у різних галузях використання. Однією з основних проблем в системах Інтернету речей є відсутність стандартів та нормативів з безпеки та захисту. Це призводить до того, що виробники пристроїв можуть застосовувати різні підходи до забезпечення безпеки та захисту своїх пристроїв, що може створювати вразливості та ризики для користувачів.

Крім того, оновлення програмного забезпечення та патчі, які містять у собі виправлення виявлених уразливостей, не завжди вчасно та ефективно поширюються серед користувачів. Ще однією проблемою є складність захисту систем Інтернету речей, оскільки вони можуть містити велику кількість різних компонентів та пристроїв, які взаємодіють між собою через бездротові мережі. Це призводить до збільшення векторів атак та зниження ефективності захисту. У зв'язку з цим, розробка та впровадження ефективних механізмів захисту в системах Інтернету речей є надзвичайно складною задачею, яка вимагає комплексного підходу. Для забезпечення безпеки систем Інтернету речей необхідно розглядати всі елементи системи як цілісну структуру та забезпечувати їх взаємодію з точки зору безпеки та захисту. Для цього потрібно розробляти та використовувати нові технології та стандарти, які дозволяють забезпечити захист від різних типів атак, а також підвищувати ефективність оновлення та патчінгу програмного забезпечення.

Мета моєї роботи полягає в розгляді механізмів захисту складових системи Інтернету речей (IoT), що дозволить зрозуміти основні загрози та вразливості, з якими стикаються при проектуванні при роботі з системами IoT, а також розробити додаток який продемонструє доречність моєї ідеї що до підвищення захисту з’єднання BlueTooth.

Основні завдання роботи

1. Дослідження основних елементів складових систем IoT;

2. Вивчення літературних джерел та наукових робіт, що описують основні елементи IoT. Детальний огляд літератури дозволить зрозуміти функціональні можливості кожного елементу, його роль у системі та способи інтеграції з іншими компонентами;

3. Розгляд принципів функціонування та взаємодії між різними елементами інтернет речей Виявлення залежностей та взаємозв'язків між системами, оцінка впливу кожного елементу на загальну функціональність системи;

4. Аналіз вразливостей і факторів у технологіях інтернету речей; 5. Розробка додатку для забезпечення безпеки з`єднання на прикладі BlueTooth.

Об'єктом дослідження є – сама система Інтернету речей (IoT)

Предметом дослідження є – механізми та методи захисту

РОЗДІЛ 1
ОГЛЯД СКЛАДОВИХ СИСТЕМИ IOT

1.1 Аналіз архітектури IoT

Архітектура Інтернету Речей (IoT) – це система, яка об’єднує різні пристрої та додатки в одну мережу, дозволяючи їм обмінюватися даними та взаємодіяти між собою. Це забезпечує підвищення ефективності та автоматизації процесів, зменшення витрат та підвищення комфорту життя. Архітектура IoT складається з наступних компонентів:

1. Сенсори та пристрої збору даних: це фізичні пристрої, які здатні зчитувати дані з навколишнього середовища, такі як температура, вологість, рух, освітлення тощо. Вони можуть відправляти ці дані до хмарного сервісу або обробляти їх локально на пристрої.

2. Хмарні сервіси: це мережі серверів, які забезпечують зберігання, обробку та аналіз даних, які зібрані з сенсорів та інших пристроїв. Хмарні сервіси можуть бути публічними, приватними або гібридними.

3. Шлюзи з’єднання: це компоненти, які забезпечують зв’язок між сенсорами та хмарними сервісами. Вони можуть обробляти та передавати дані від сенсорів до хмарного сервісу або навпаки, забезпечуючи шифрування та захист від несанкціонованого доступу.

4. Програмне забезпечення: це компоненти, які дозволяють керувати та моніторити IoT-систему. Вони можуть бути вбудованими в пристрої та сенсори, або мати вигляд окремих додатків, які можуть бути встановлені на різних пристроях.

Ці компоненти співпрацюють між собою, щоб забезпечити плавну роботу мережі IoT. Дані, що збираються сенсорами, можуть бути передані до хмарного сервісу для аналізу та обробки, щоб отримати корисну інформацію. За допомогою програмного забезпечення, користувачі можуть контролювати та керувати своїми IoT-пристроями, отримувати повідомлення та аналізувати дані [1].

Архітектура IoT зазвичай складається з декількох рівнів. Найнижчим рівнем є рівень датчиків та пристроїв збору даних, що збирають інформацію про навколишнє середовище, стан обладнання та інші параметри. Ці дані потім передаються до рівня мережі, де вони обробляються, аналізуються та передаються до хмарного сервісу для подальшої обробки та зберігання. На рівні мережі можуть бути використані різні технології передачі даних, такі як Wi-Fi, BlueTooth, ZigBee, NFC та інші. Важливо забезпечити безпеку передачі даних на цьому рівні, використовуючи шифрування та інші методи захисту інформації. Вище за рівнем мережі зазвичай знаходиться рівень платформи, де дані можуть бути оброблені та аналізовані. Цей рівень може включати в себе різні компоненти, такі як бази даних, сервіси аналізу даних, веб-сервери та інші. Останнім рівнем є рівень застосування, де дані можуть використовуватися для різних цілей, таких як контроль за обладнанням, моніторинг стану систем, автоматизація процесів та інше. Для роботи на цьому рівні можуть бути використані різні програмні засоби, такі як мобільні додатки, веб-інтерфейси, програмне забезпечення для керування системами та інше [2].

Проте, однією з основних проблем IoT є безпека. Збільшення кількості підключених пристроїв та взаємодія між ними створює більше можливостей для кібератак та порушення безпеки. Тому важливо вживати заходів для захисту системи IoT, таких як шифрування даних, автентифікація користувачів, забезпечення безпеки мереж, підключення пристроїв та інше. Однією з найважливіших переваг IoT є здатність до збору та аналізу великої кількості даних з різних джерел. Це дозволяє розробникам розумних пристроїв та систем розробляти нові технології та рішення для покращення життя людей.

У медицині, наприклад, IoT може допомогти в покращенні діагностики та лікування захворювань, а також нагляду за пацієнтами. Розумні медичні пристрої можуть збирати та передавати дані про стан здоров’я пацієнтів до медичних систем, що дозволяє швидше та точніше діагностувати та лікувати різні захворювання. У транспорті IoT може допомогти в покращенні безпеки та ефективності руху транспорту. Розумні системи допомагають збирати та аналізувати дані про рух транспорту, що дозволяє покращити регулювання руху та зменшити кількість аварій.

У промисловості IoT може допомогти в автоматизації та оптимізації виробничих

процесів. Розумні системи контролю та управління можуть збирати та аналізувати дані про виробничі процеси, що дозволяє покращити їх ефективність та зменшити витрати на виробництво. Однак, разом з перевагами IoT, існують і певні ризики. Збільшення кількості підключених до мережі пристроїв створює більше можливостей для кібератак та порушення безпеки. Тому важливо вживати заходів для захисту системи IoT, таких як шифрування даних, автентифікація користувачів та забезпечення безпеки мережі.

Окрім вищенаведених відомих переваг та ризиків, IoT має також значний вплив на розвиток економіки та суспільства. За допомогою IoT можна створювати нові бізнес-моделі та послуги, що дозволяє підприємствам збільшувати ефективність виробництва та покращувати взаємодію з клієнтами. Наприклад, віддалене моніторингові та сервісні системи можуть допомогти підприємствам відслідковувати стан своїх пристроїв та устаткування, планувати їхнє технічне обслуговування та відновлення, а також збирати дані для покращення якості продукції та процесів виробництва. Іншим прикладом є розумні будинки та міста, які можуть забезпечити зручність та безпеку для жителів, а також допомогти знизити витрати на енергію та водопостачання. Розумні системи контролю та управління можуть автоматично регулювати енергоспоживання та опалення в будинках, що зменшує витрати на комунальні послуги та сприяє збереженню енергоресурсів. У соціальній сфері IoT може допомогти забезпечити безпеку та зручність для людей з різними видами обмежень та потребами. Наприклад, розумні пристрої можуть допомогти людям з обмеженнями у русі або зорі здійснювати різні дії, такі як відкривання дверей або управління освітленням, за допомогою голосових команд або жестів. Крім того, IoT може допомогти вирішити проблеми в сфері охорони здоров’я. Розумні медичні пристрої можуть забезпечувати постійний моніторинг стану здоров’я пацієнтів та передавати дані про цей стан до лікарів. Це може допомогти лікарям вчасно виявляти стан погіршення та призначати необхідні лікувальні процедури.

Крім того, розумні медичні пристрої можуть допомогти пацієнтам віддалено здійснювати консультації з лікарями, отримувати рекомендації щодо свого стану здоров’я та здійснювати необхідні дії, такі як прийом ліків або вимірювання показників здоров’я. Незважаючи на всі переваги, IoT також має свої ризики. Збір та

збереження великих обсягів даних, що генеруються IoT-пристроями, може призвести до проблем з конфіденційністю та захистом даних користувачів. Крім того, недостатня захищеність IoT-пристроїв може спричинити їх вразливість до кібератак та зловживань, що може призвести до порушення функціонування систем та потенційно небезпечних ситуацій. Отже, розвиток IoT має великий потенціал для покращення якості життя та ефективності виробництва, однак його використання повинно бути супроводжуватися відповідними заходами захисту приватності та кібербезпеки [3].

1.2 Вивчення методів забезпечення конфіденційності інформації

Інтернет речей (IoT) складається з різних компонентів, які співпрацюють для створення підключеної системи. Основними складовими системи IoT є: • Під’єднані пристрої: Це фізичні пристрої або сенсори, які здатні збирати та передавати дані через Інтернет. Наприклад, це можуть бути датчики вимірювання температури, вологості, руху або вимикачі, які з'єднуються з мережею для передачі інформації.

• Засоби збору даних: Це компоненти, які забезпечують збір, агрегацію та обробку даних, зібраних від підключених пристроїв. Вони можуть включати хмарні платформи, локальні сервери або вбудовані системи збору даних.

• Комунікаційні мережі: Це мережеві засоби, що забезпечують передачу даних між підключеними пристроями та засобами збору даних. Це можуть бути мережі Wi Fi, BlueTooth, ZigBee, Z-Wave, LTE, NB-IoT та інші.

• Хмарні сервіси: Хмарні платформи надають обчислювальні та зберігаючи ресурси для зберігання, обробки та аналізу великого обсягу даних, зібраних від підключених пристроїв. Вони також можуть надавати інструменти для розробки додатків IoT та управління системою.

• Аналітика та штучний інтелект: Ці компоненти забезпечують аналіз та витяг інформації з великого обсягу даних, що зібрані від підключених пристроїв. Вони можуть використовувати алгоритми машинного навчання та інші техніки штучного інтелекту для отримання цінних відомостей та прогнозування майбутніх подій. Наприклад, аналіз даних може допомогти прогнозувати технічні проблеми з

пристроями до їх виникнення, а також забезпечити оптимізацію енергоспоживання та забезпечити високу ефективність системи IoT.

1. Системи управління: Це компоненти, що забезпечують управління підключеними пристроями та додатками IoT. Вони можуть надавати інтерфейси для моніторингу та керування пристроями з використанням мобільних додатків або веб інтерфейсів.

2. Безпека: Це компоненти, що забезпечують захист підключених пристроїв та даних від зловмисних атак. Вони можуть включати криптографічні протоколи, віддалені аутентифікаційні методи та інші методи захисту.

Для забезпечення ефективної роботи системи IoT, всі ці компоненти мають бути добре інтегровані між собою. Наприклад, підключені пристрої повинні бути сумісні з протоколами мережевої взаємодії, щоб забезпечити безперебійний обмін даними зі спеціалізованими платформами для збору та аналізу даних. Крім того, компоненти системи IoT мають забезпечувати масштабованість, щоб забезпечити роботу системи при збільшенні обсягів даних та підключених пристроїв. У світі IoT, стандарти відіграють важливу роль у забезпеченні взаємодії між різними пристроями та системами. Наприклад, стандарт Wi-Fi дозволяє підключати бездротові пристрої до мережі Інтернет, стандарт BlueTooth дозволяє підключати пристрої в межах невеликих відстаней, а стандарт ZigBee дозволяє підключати пристрої до великих мереж з високою енергоефективністю. У майбутньому, системи IoT будуть ставити нові завдання перед компонентами системи, такими як збільшення швидкості передачі даних, забезпечення максимальної безпеки, підтримка нових протоколів мережевої взаємодії та забезпечення масштабованості [4].

З розвитком технологій та збільшенням популярності IoT, очікується, що системи IoT будуть забезпечувати нові можливості для підприємств та споживачів, забезпечуючи нові рівні ефективності та зручності. Одним з важливих аспектів систем IoT є забезпечення безпеки даних та пристроїв.

Оскільки пристрої IoT містять багато конфіденційної інформації про користувачів та їхні звички, вони можуть стати об'єктом кібератак, які можуть призвести до витоку даних або пошкодження пристроїв. Для забезпечення безпеки даних та пристроїв у системах IoT використовуються різноманітні технології, такі як

шифрування, аутентифікація, контроль доступу та інші. Ще одним важливим аспектом систем IoT є забезпечення енергоефективності. Більшість пристроїв IoT працюють від батареї або енергозберігаючих джерел енергії, тому важливо забезпечити мінімальне споживання енергії, щоб забезпечити тривалу автономну роботу. Для цього використовуються різноманітні технології, такі як оптимізація роботи пристроїв, зменшення частоти передачі даних та використання спеціальних протоколів мережевої взаємодії. Ще одним важливим аспектом систем IoT є збір та аналіз даних. Оскільки пристрої IoT збирають великі обсяги даних, їхнє аналіз та інтерпретація може забезпечити корисну інформацію для підприємств та організацій [5].

Для забезпечення ефективного збору та аналізу даних використовуються різноманітні інструменти та платформи, такі як хмарні сервіси, інструменти машинного навчання та аналізу даних. Однак, на жаль, у систем IoT також є певні недоліки. Наприклад, вони можуть бути вразливі до кібератак, які можуть призвести до витоку конфіденційної інформації та пошкодження пристроїв. Крім того, питання безпеки даних у системах IoT ще не повністю вирішені, і вони залишаються предметом дослідження та вдосконалення. Одним з ключових напрямків застосування систем IoT є індустрія 4.0.

Це концепція "розумної фабрики", яка передбачає використання мережі підключених пристроїв для підвищення ефективності виробничих процесів. В рамках цієї концепції, машини та пристрої взаємодіють між собою та з операторами в режимі реального часу, що дозволяє підвищити продуктивність та знизити витрати на виробництво. Ще одним важливим напрямком застосування систем IoT є "розумні міста" [6].

У рамках цієї концепції, міста використовують мережу підключених пристроїв для забезпечення ефективної роботи інфраструктури, підвищення безпеки та комфорту громадян. Такі системи можуть включати в себе розумне освітлення, системи моніторингу трафіку, автоматизовану парковку, а також системи контролю якості повітря та води. Також системи IoT знаходять своє застосування у сфері здоров'я та фітнесу. Завдяки підключеним пристроям, люди можуть вести моніторинг свого стану здоров'я, отримувати рекомендації щодо здорового способу життя та

взаємодіяти зі своїми лікарями в режимі реального часу.

Окрім цього, системи IoT можуть бути корисними у сфері енергетики та екології. Завдяки підключеним пристроям, можливо зменшити витрати на енергоспоживання та покращити екологічну ситуацію. Наприклад, системи IoT можуть використовуватися для моніторингу та контролю якості повітря, води та ґрунту. Важливим елементом систем IoT є збір та обробка великої кількості даних. Для цього використовуються різноманітні алгоритми машинного навчання та інші методи аналізу даних, які дозволяють отримати цінні інсайти та покращити функціональні можливості систем IoT. Одним з головних викликів, який виникає при розробці систем IoT, є забезпечення безпеки даних. Тому розробники систем IoT повинні забезпечувати найвищий рівень безпеки та захисту даних, що передбачає використання різноманітних методів шифрування, ідентифікації та аутентифікації.

Однією з головних переваг систем IoT є їх можливість забезпечувати автоматизоване та дистанційне управління різними процесами та пристроями. Наприклад, в сфері промисловості системи IoT використовуються для моніторингу технічного стану обладнання та забезпечення його ефективної роботи, а також для контролю за виробничими процесами та забезпечення безпеки на робочому місці. У сфері будівництва та управління містами системи IoT можуть використовуватися для моніторингу технічного стану будівель та інфраструктури, контролю за енергоспоживанням та оптимізації роботи систем опалення, вентиляції та кондиціонування повітря. Також можливе використання систем IoT для контролю транспортного потоку, покращення роботи публічного транспорту та забезпечення безпеки на дорогах. Однією з найбільш важливих можливостей систем IoT є їх здатність взаємодіяти з різноманітними пристроями та системами. Це дозволяє створювати інноваційні рішення та інтегровані системи, які дозволяють покращувати ефективність та комфорт життя людей.

Одним з переваг систем IoT є їх економічна вигода. За допомогою систем IoT можна зменшити витрати на енергоспоживання та обслуговування обладнання, збільшити ефективність виробництва та управління містами, а також забезпечити збільшення прибутку підприємств. Щоб зменшити ризики та забезпечити безпеку використання систем IoT, необхідно впроваджувати заходи забезпечення

інформаційної безпеки та здійснювати моніторинг роботи систем на постійній основі. Для цього можуть використовуватися спеціальні програмні засоби та алгоритми, які дозволяють виявляти та запобігати кібератакам. Також важливо зазначити, що системи IoT мають великий потенціал у сфері медицини та охорони здоров'я. Наприклад, можливо створення медичних пристроїв, які забезпечують постійний моніторинг стану пацієнтів та автоматично реагують на можливі небезпеки. Такі пристрої можуть бути особливо корисні для пацієнтів з хронічними захворюваннями та потребують постійної медичної допомоги. Додатково, системи IoT можуть використовуватися для покращення енергоефективності та екологічної стійкості. Наприклад, за допомогою датчиків та розумних систем управління можна зменшити споживання електроенергії та води, знизити викиди шкідливих речовин та підвищити ефективність виробництва. Узагалі, системи IoT є перспективним напрямком розвитку технологій та мають великий потенціал у різних сферах життя. Однак, для того, щоб цей потенціал було реалізовано на повну міру, необхідно забезпечувати безпеку та конфіденційність даних, а також враховувати можливі ризики та використовувати відповідні заходи захисту від кібератак та злочинних дій.

1.3 Дослідження принципів функціонування кожної складової системи

Інтернет речей (IoT) – це мережа фізичних пристроїв, які підключені до Інтернету і можуть обмінюватися даними між собою та з іншими системами. IoT складається з трьох основних складових: пристроїв, мереж та хмарної інфраструктури. Описуючи принципи функціонування кожної з цих складових, ми можемо отримати наступне:

1. Пристрої: Пристрої IoT – це фізичні пристрої, такі як датчики, реле, контролери тощо, які можуть зчитувати дані з навколишнього середовища, відправляти дані в мережу, а також приймати команди з хмари. Основним принципом функціонування пристроїв IoT є збір, обробка та передача даних. Для цього пристрої повинні мати вбудовані датчики та засоби збору даних, а також підтримувати мережеві протоколи для обміну даними з іншими пристроями та хмарним сервісом.

2. Мережі: Мережі IoT – це мережеві інфраструктури, які забезпечують зв’язок

між пристроями та іншими мережами. Основним принципом функціонування мереж IoT є передача даних з пристроїв до хмарної інфраструктури, а також передача команд з хмарної інфраструктури до пристроїв. Для цього мережі повинні мати високу пропускну здатність, надійність та можливість взаємодії з іншими мережами та сервісами.

3. Хмарна інфраструктура: Хмарна інфраструктура IoT – це сервери та сервіси, які забезпечують зберігання, обробку та аналіз даних, які надходять з пристроїв та мереж. Основним принципом функціонування хмарної інфраструктури IoT є забезпечення підтримки масштабування, безпеки та доступності. Для цього хмарна інфраструктура повинна мати велику потужність обчислень, високий рівень захисту даних та можливість працювати з різними пристроями та мережами.

Усі складові IoT повинні працювати разом, щоб забезпечити ефективну та безпечну обробку даних. Пристрої збирають дані та передають їх до мереж, які передають дані до хмарної інфраструктури, де вони аналізуються та зберігаються. Потім результати аналізу можуть бути відправлені назад до пристроїв або використані для прийняття рішень. Для успішної роботи системи IoT необхідна ефективна інтеграція між пристроями, мережами та хмарною інфраструктурою. Кожна складова повинна бути розроблена та налаштована з урахуванням специфіки роботи у мережі IoT та забезпечувати безпеку та конфіденційність обміну даними. Пристрої повинні бути здатні до збору, передачі та обробки даних в режимі реального часу.

Для цього вони повинні мати підтримку різних мережевих технологій та протоколів, таких як Wi-Fi, BlueTooth, ZigBee та інших. Крім того, вони повинні мати вбудовані датчики та сенсори, які дозволяють збирати інформацію про різні параметри навколишнього середовища, такі як температура, вологість, освітленість, тиск та інші. Мережі IoT повинні бути розгорнуті з урахуванням потреб користувачів та мають бути здатні до передачі великої кількості даних з високою швидкістю та надійністю. Це можливо завдяки застосуванню різних технологій та протоколів мережевого зв’язку, таких як LoRaWAN, NB-IoT, LTE-M та інші. Хмарна інфраструктура IoT повинна забезпечувати потужність обчислень та міцність зберігання даних, а також високий рівень захисту та конфіденційності даних. Для

цього можуть використовуватися різні технології та рішення, такі як хмарні платформи, системи аналізу даних, блокчейн технології та інші. Крім того, система IoT повинна забезпечувати високий рівень безпеки та конфіденційності, що дозволяє запобігати несанкціонованому доступу до даних та захищати користувачів від ризиків кібератак та крадіжки даних. Для цього можуть використовуватися різні методи та технології, такі як шифрування даних, аутентифікація користувачів, системи ідентифікації та авторизації та інші. Однією з ключових складових системи IoT є датчики та сенсори. Вони забезпечують збір даних про різні параметри навколишнього середовища та процесів, що відбуваються у системі. Ці дані можуть використовуватися для аналізу та оптимізації роботи системи, забезпечення її безпеки та енергоефективності. Іншою складовою системи IoT є мережеві пристрої та комунікаційні протоколи, які дозволяють об'єднати всі складові системи в єдину мережу. Це можуть бути роутери, маршрутизатори, мережеві гейти та інші пристрої, які забезпечують безперервну передачу даних між пристроями та системами [7].

Іншою важливою складовою є облікові записи та системи керування, які дозволяють користувачам взаємодіяти з системою та керувати її різними параметрами. Наприклад, користувачі можуть встановлювати параметри системи, контролювати її роботу, отримувати повідомлення про стан системи та інше. Також до складових системи IoT можуть входити хмарні технології, що дозволяють зберігати та обробляти великі обсяги даних в режимі реального часу. Це можуть бути облікові записи, сервіси аналізу даних, сервіси забезпечення безпеки та інші. Додатково слід зазначити, що системи IoT можуть бути різних розмірів та складності. Від малих систем домашньої автоматизації, що включають у себе кілька пристроїв, до великих систем, що об'єднують сотні тисяч пристроїв та мільйони користувачів по всьому світу. Важливим аспектом розвитку систем IoT є стандартизація та забезпечення сумісності між різними пристроями та системами. Це дозволяє забезпечити взаємодію між різними системами, а також сприяти розвитку інтернету речей як цілісної галузі. Одним з викликів для систем IoT є захист від кібератак та забезпечення приватності даних.

З огляду на те, що системи IoT збирають та оброблюють велику кількість даних, необхідно забезпечити надійний захист від зловмисників та несанкціонованого

доступу до інформації. Також слід враховувати питання приватності користувачів та дотримання відповідних законодавчих вимог щодо обробки та збереження даних. Ще одним важливим аспектом розвитку систем IoT є їх енергоефективність. З огляду на те, що багато пристроїв IoT мають обмежені ресурси енергопостачання, необхідно забезпечити ефективне використання енергії та зменшення споживання. Це може бути досягнуто за допомогою оптимізації процесів, зменшення частоти відправки даних, використання спеціалізованих пристроїв з енергозберігаючими технологіями та ін.

Крім того, системи IoT дозволяють забезпечувати більш точне та ефективне управління різними процесами та системами. Наприклад, за допомогою системи автоматичного контролю вологості та температури, можна забезпечити оптимальні умови для зберігання продуктів харчування, що дозволяє зменшити втрати та зберегти якість продуктів. Автоматизовані системи управління електричною мережею дозволяють забезпечити більш ефективне використання енергії та зменшити витрати на її постачання. Для забезпечення захисту системи IoT використовуються різні заходи та технології. Одним з найважливіших елементів є захист мережі та обладнання від несанкціонованого доступу. Для цього можуть використовуватись різні методи, такі як шифрування трафіку, використання паролів та ідентифікаторів, фільтрація мережевого трафіку та багато інших. Крім того, можуть використовуватись системи пошуку та захисту від зловмисних програм, а також механізми для виявлення та реагування на кібератаки. Ще одним важливим аспектом захисту системи IoT є забезпечення конфіденційності та цілісності даних, які обробляються та передаються через систему [8].

Для цього використовуються різні методи шифрування та контролю цілісності даних, що забезпечують їх захист від несанкціонованого доступу та змін. Нарешті, важливим аспектом захисту системи IoT є забезпечення доступності сервісів та даних, які надаються через систему. Для цього використовуються методи балансування навантаження, резервного копіювання та відновлення після аварійних ситуацій. У кінці кінців, забезпечення безпеки системи IoT є складним завданням, яке вимагає комплексного підходу та використання різних методів та технологій. З метою забезпечення ефективної та надійної роботи системи IoT важливо враховувати всі аспекти безпеки, включаючи фізичну безпеку, захист мережі, захист даних та

додатків, оновлення та виправлення вразливостей, аутентифікацію та авторизацію користувачів, а також балансування навантаження та резервне копіювання даних [9].

РОЗДІЛ 2
ЗАГРОЗИ БЕЗПЕЦІ СКЛАДОВИХ СИСТЕМИ ІОТ

2.1 Класифікація загроз за походженням та способами реалізації

Системи Інтернету речей (loT) можуть бути піддані різноманітним загрозам, що можуть походити з різних джерел та мати різні способи реалізації. Розглянемо деякі класифікації загроз за їх походженням та способами реалізації:

1. За походженням:

• Внутрішні загрози: це загрози, що виникають внаслідок дій працівників компанії, які мають доступ до системи loT. Такі загрози можуть включати в себе зловживання привілеями, неправильне використання даних або навмисне викривлення функцій системи.

• Зовнішні загрози: це загрози, що походять з-за меж системи loT. Такі загрози можуть включати в себе хакерські атаки, фішингові атаки, віруси та інші. 2. За способами реалізації:

• Атаки на безпеку: це загрози, що використовуються для зламування системи loT і отримання неправомірного доступу до неї. Такі загрози можуть включати в себе перехоплення даних, внесення змін у функціонування системи, розголошення конфіденційних даних тощо.

• Атаки на конфіденційність: це загрози, що використовуються для порушення приватності користувачів системи loT. Такі загрози можуть включати в себе відстеження розташування, перехоплення повідомлень, збір інформації про користувача тощо.

• Атаки на доступність: це загрози, що використовуються для перешкоджання нормальному функціонуванню системи loT. Такі загрози можуть включати в себе DDoS-атаки, внесення змін у налаштування системи тощо. 3. За типом пристроїв loT:

• Загрози для домашніх пристроїв loT: це загрози, що виникають через підключення домашніх пристроїв до Інтернету, таких як роутери, маршрутизатори,

домашні системи безпеки тощо. Такі пристрої можуть стати ціллю атак з-за їх вразливостей та недостатньої захищеності.

• Загрози для промислових пристроїв IoT: це загрози, що виникають внаслідок підключення промислових пристроїв до Інтернету, таких як датчики виробничого процесу, медичне обладнання, системи контролю доступу тощо. Такі пристрої можуть бути ціллю спроб хакерських атак або використовуватися як точки входу для зламування інших систем в мережі.

4. За масштабом впливу:

• Локальні загрози: це загрози, що впливають на окремі пристрої IoT або на малі групи пристроїв, такі як домашні мережі.

• Глобальні загрози: це загрози, що мають широкомасштабний вплив на системи IoT, такі як масштабні хакерські атаки на критичну інфраструктуру мережі або на системи управління енергопостачанням та транспортом.

Крім того, загрози для систем IoT можуть виникати на різних етапах використання системи, таких як:

1. Етап розробки: на цьому етапі можуть виникати загрози через недостатню увагу до безпеки при розробці програмного забезпечення та жорсткого забезпечення, недостатню перевірку на вразливості та недостатній захист від атак.

2. Етап експлуатації: на цьому етапі загрози можуть виникати через недостатній рівень захисту та недостатню оновлення програмного забезпечення та жорсткого забезпечення.

3. Етап знищення: на цьому етапі можуть виникати загрози через недостатній рівень захисту від несанкціонованого доступу до даних та можливості відновлення даних після знищення пристрою.

Однією з найбільш поширених загроз для систем IoT є атаки на мережевий рівень. Ці атаки можуть включати в себе різні методи, такі як перехоплення трафіку, внесення змін у пакети даних, введення підроблених пакетів у мережу, відправку фальшивих запитів та багато іншого. Іншим типом загроз для систем IoT є атаки на рівень додатків. Ці атаки можуть включати в себе експлойти, злом програмного забезпечення, фішингові атаки та інші методи. Зокрема, фішингові атаки є особливо

небезпечними для систем loT, оскільки вони можуть бути виконані шляхом відправки шахрайських повідомлень, які виглядають як легітимні запити до системи. Якщо користувач системи повірить цим повідомленням і введе свої дані, то зловмисник може отримати доступ до системи та виконувати дії, які можуть завдати шкоди. Крім того, іншим типом загроз для систем loT є атаки на рівень жорсткого забезпечення.

Ці атаки можуть включати в себе експлойти, які використовують вразливості у жорсткому забезпеченні, техніки реверс-інжинірингу, фізичний доступ до пристрою та інші методи. Усі ці загрози можуть мати серйозні наслідки для підприємств та організацій, які використовують системи loT.

Тому для захисту від цих загроз необхідно вживати різноманітні заходи забезпечення безпеки, такі як шифрування даних, перевірка на вразливості, мережеві заходи безпеки та інші. Також важливо регулярно оновлювати програмне забезпечення та жорстке забезпечення системи та враховувати нові загрози, які можуть з'являтися з часом. Зокрема, оновлення програмного забезпечення та жорсткого забезпечення можуть включати в себе виправлення вразливостей та захист від нових загроз. Крім того, важливо регулярно навчати та підвищувати кваліфікацію персоналу, який відповідає за безпеку системи loT [10].

Це може включати в себе проведення тренінгів та семінарів, які допоможуть працівникам підприємства розуміти нові загрози та вміти ефективно захищати систему. Також, для захисту від загроз системи loT можуть використовуватися різні технології та підходи, такі як блокчейн, машинне навчання, штучний інтелект та інші. Застосування цих технологій може допомогти підприємствам ефективно боротися зі загрозами та забезпечувати високий рівень безпеки системи loT. Для захисту системи loT важливо також використовувати механізми автентифікації та авторизації, які дозволяють перевіряти, що пристрій, який намагається отримати доступ до системи, дійсно має право на цей доступ. Це може включати в себе використання паролів, двофакторної аутентифікації, біометричних методів та інших методів перевірки. Також важливо забезпечити захист даних, які передаються та зберігаються в системі loT. Це може включати в себе шифрування даних, контроль доступу до них та

забезпечення їх безпеки під час зберігання та передачі. Окрім технічних заходів, важливо також враховувати правові та регуляторні аспекти захисту системи loT.

Зокрема, підприємства повинні дотримуватися законодавства щодо захисту персональних даних та інших регуляторних вимог, які стосуються захисту системи IoT. Загалом, захист системи IoT є складним завданням, яке потребує використання комплексного підходу та застосування різноманітних технічних та неприкметних заходів. Захист системи IoT не тільки забезпечує безпеку підприємства, але також допомагає захистити від різноманітних кібератак та злочинів, які можуть стати на шляху до ефективного функціонування системи.

Для ефективного захисту системи IoT також важливо використовувати моніторинг та аналітику даних. Це дозволяє виявляти аномальні показники роботи системи та негайно реагувати на можливі загрози безпеки. Наприклад, виявлення надмірної активності певного пристрою в системі IoT може бути індикатором того, що пристрій був скомпрометований та його використовують для злочинних цілей. Додатково, важливо мати плани навчання та свідомості для користувачів та робітників, які мають доступ до системи IoT. Це включає в себе навчання щодо безпечного використання та збереження даних, використання безпечних паролів та інших методів захисту, а також виявлення та повідомлення про можливі загрози безпеки.

Нарешті, важливо забезпечити постійну оновлення програмного забезпечення та апаратного забезпечення в системі IoT. Це дозволяє запобігти використанню застарілих методів захисту та вразливостей, які можуть бути використані хакерами для зламу системи. Розробники системи повинні постійно слідкувати за новими загрозами та оновлювати систему, щоб забезпечити її безпеку. Взагалі, захист системи IoT є важливою складовою її ефективного та безпечного функціонування. Застосування комплексного підходу до захисту системи, який включає в себе технічні, правові та неприкметні заходи, дозволяє забезпечити захист від різноманітних загроз безпеки та зберегти функціональність та ефективність системи IoT.

2.2 Розгляд відомих прикладів атак на системи IoT

Інтернет речей (IoT) охоплює велику кількість пристроїв, що пов’язані з Інтернетом, включаючи домашні пристрої, автомобілі, медичні прилади та інше. Оскільки більшість з цих пристроїв мають обмежені ресурси, вони зазвичай не мають вбудованих механізмів захисту, що робить їх вразливими до атак. Ось деякі з відомих прикладів атак на системи IoT:

1. Mirai: у 2016 році було виявлено ботнет Mirai, який використовувався для здійснення атак DDoS. Цей ботнет використовував вразливості в пристроях IoT, таких як маршрутизатори, камери відеоспостереження та DVR, для створення ботнету.

2. BlueBorne: ця атака використовувала вразливості в протоколах BlueTooth, що дозволяли зловмисникам віддалено виконувати код на пристроях IoT, таких як смартфони, навушники та інші пристрої, що підтримують BlueTooth.

3. Wannacry: хоча ця атака була спрямована на комп’ютери, вона також може вплинути на пристрої IoT, які працюють під управлінням Windows. Атака використовувала вразливість в операційній системі Windows, що дозволяла зловмисникам віддалено виконувати код на комп’ютерах та пристроях IoT.

4. BrickerBot: ця атака використовувала вразливості в пристроях IoT, що призвело до того, що пристрої були назавжди пошкоджені. Замість того, щоб використовувати пристрої для здійснення атак, BrickerBot використовував вразливості, щоб знищити пристрої.

5. Stuxnet: ця атака була спрямована на комп’ютери, але вона також мала вплив на промислові системи IoT, зокрема на системи керування промисловими процесами. Stuxnet використовував вразливості в операційній системі Windows та контролерів програмованої логіки (PLC), які використовувалися в іранському ядерному заводі, для віддаленого керування промисловими процесами та зниження продуктивності. Ця атака показала, що промислові системи IoT можуть бути цілком вразливі до кібератак, що може мати серйозні наслідки для безпеки та економічної стійкості підприємств.

Узагалі, кібербезпека є серйозною проблемою для систем IoT, тому що більшість пристроїв не має ефективних механізмів захисту та легко стають метою для кіберзлочинців. Щоб зменшити ризики, пов’язані з кібератаками, необхідно приділяти

більшу увагу кібербезпеці при розробці та експлуатації пристроїв IoT, використовувати захист від DdoS атак, шифрування даних та вчасне оновлення програмного забезпечення. Додатковим прикладом атак на системи IoT є Mirai, ботнет, який був відповідальний за одну з найбільших DdoS атак в історії Інтернету. Mirai використовував вразливості в безпеці багатьох пристроїв IoT, таких як маршрутизатори, камери спостереження та домашній принтери, для створення мережі зі зброєю зомбі. Ця мережа зі зброєю зомбі була потім використана для атаки на веб-сайти та інші цифрові послуги. Іншим прикладом атак на системи IoT є BlueBorne, вразливість, що дозволяє зловмисникам віддалено взяти під контроль пристрої з BlueTooth. BlueBorne може бути використана для крадіжки даних, виконання шкідливого коду, а також для створення мереж зі зброєю зомбі. Окрім вищезгаданих прикладів атак на системи IoT, існують й інші типи кібератак, які можуть стати небезпечними для цих систем. Наприклад, атака на систему IoT може бути проведена шляхом використання фішингу, коли зловмисники намагаються переконати користувачів відкрити шкідливі посилання або завантажити шкідливі файли. Також можуть бути використані атаки на протоколи комунікації, такі як DNS або BGP, для зміни маршрутів комунікації та перехоплення даних. У такому випадку зловмисники можуть використовувати зламані пристрої IoT як проміжні вузли для перехоплення даних або для зміни маршруту комунікації. Іншим видом атак на системи IoT є атаки на безпеку мережі, такі як MITM (Man-in-the-Middle) атака, де зловмисники перехоплюють комунікацію між двома пристроями, що знаходяться на різних кінцях мережі, і можуть змінювати передану інформацію або крадіжку даних.

Ще однією загрозою для систем IoT є DdoS-атаки, коли зловмисники використовують багато пристроїв IoT, які були зламані і взяті під контроль, для створення ботнету і нападу на мережевий пристрій або сервер. В результаті цього може статися перевантаження мережі, що може призвести до відмови в роботі системи. Також, важливим аспектом безпеки систем IoT є захист від зламування паролів [11].

Часто зловмисники намагаються зламати паролі, щоб отримати доступ до пристроїв IoT, які мають вразливість до атак, і використовують їх для своїх злочинних дій. Крім того, багато пристроїв IoT використовують старі, застарілі версії

програмного забезпечення, які мають вразливості до атак. Тому важливо регулярно оновлювати програмне забезпечення, щоб запобігти використанню старих вразливих версій. Загалом, безпека систем IoT є важливим питанням, яке вимагає постійної уваги і підходу до проблеми з усіх сторін. Крім вживання заходів забезпечення безпеки, таких як шифрування даних і регулярне оновлення програмного забезпечення, важливо також вести моніторинг мережі та пристроїв, щоб виявляти будь-які ознаки атак і приймати вчасні заходи щодо їх запобігання. Ще однією потенційною загрозою для систем IoT є атаки з використанням зловмисних програм.

Наприклад, зловмисники можуть створювати шкідливі програми, які можуть встановлюватися на пристрої IoT через вразливості в програмному забезпеченні або через використання слабких паролів. Ці програми можуть дозволити зловмисникам отримати несанкціонований доступ до системи, отримувати конфіденційну інформацію або виконувати інші злочинні дії. Ще однією загрозою є атаки з використанням фізичного доступу до пристроїв IoT. Зловмисники можуть отримати фізичний доступ до пристрою, щоб змінити його настройки або встановити шкідливе програмне забезпечення.

Наприклад, зловмисники можуть підключити собі до пристрою, використовуючи USB-порт або інші інтерфейси, щоб віддалено взяти його під контроль або використовувати для інших злочинних дій. Також, важливим аспектом безпеки систем IoT є захист від атак на комунікаційний протокол, що використовується між пристроями IoT. Зловмисники можуть використовувати вразливості в протоколах, щоб перехоплювати та змінювати дані, що передаються між пристроями IoT, або використовувати ці вразливості для отримання доступу до системи. Також, важливою складовою безпеки є моніторинг та аналіз поведінки системи IoT. Компанії повинні встановлювати механізми моніторингу трафіку та виявлення аномальних дій, щоб вчасно реагувати на потенційні загрози. Додатково до заходів забезпечення безпеки, про які я вже згадував, існує кілька додаткових підходів до захисту систем IoT від зловмисних атак. Один з таких підходів – це застосування технології блокчейн. Блокчейн – це розподілена база даних, яка зберігає інформацію в блоках, які підключаються один до одного ланцюжком. Ця технологія забезпечує високий рівень безпеки та відстежування даних, що є важливим для систем IoT, які

передають важливі дані. Інший підхід – це застосування штучного інтелекту [12].

Штучний інтелект може виявляти аномальні дії та виробляти прийняття рішень у реальному часі для запобігання атакам. Наприклад, система може використовувати машинне навчання для виявлення незвичайних дій у поведінці підключених пристроїв, що дозволить швидко виявити потенційні загрози та зупинити їх поширення. Крім того, деякі компанії використовують технологію кіберфізичних систем (CPS), яка поєднує в собі аспекти фізичних та кібернетичних систем для покращення безпеки та ефективності. Застосування CPS може дозволити компаніям забезпечувати більш ефективний моніторинг та контроль над підключеними пристроями, зменшуючи ризик виникнення кібератак. Усі ці заходи можуть бути ефективними для захисту систем IoT від зловмисних атак. Проте, важливо пам’ятати, що кожна система є унікальною, і потребує індивідуального підходу до захисту від кібератак.

Враховуючи все вищезазначене, а також слідкуючи за останніми тенденціями в кібербезпеці, компанії можуть забезпечити безпеку своїх систем IoT та захистити себе від можливих кібератак.

РОЗДІЛ 3
МЕХАНІЗМИ ЗАХИСТУ СКЛАДОВИХ СИСТЕМИ IOT

3.1 Методи аутентифікації та авторизації

Аутентифікація та авторизація - це дві важливі концепції, що використовуються в інформаційній безпеці для забезпечення доступу до різних ресурсів. Аутентифікація відноситься до процесу перевірки того, що користувач є тим, за кого він себе видає, а авторизація відноситься до процесу надання прав доступу користувачеві до певного ресурсу [13].

Існує кілька методів аутентифікації, таких як:

1. Логін та пароль: це найбільш поширений метод, коли користувач повинен ввести свій логін та пароль для доступу до ресурсу.

2. Біометрична аутентифікація: цей метод використовує фізичні характеристики користувача, такі як відбитки пальців, розпізнавання обличчя, розпізнавання голосу і т.д.

3. Аутентифікація на основі сертифікатів: в цьому методі користувач отримує сертифікат, який містить інформацію про його ідентичність. Користувач вводить сертифікат для аутентифікації.

Методи авторизації також можуть розглядатися як способи надання прав доступу користувачеві. Основні методи авторизації включають наступні: 1. Ролева авторизація: в цьому методі користувачам присвоюється роль, яка визначає їх рівень доступу до ресурсу.

2. Авторизація на основі дозволів: цей метод використовує набір дозволів, які надають користувачеві право доступу до певного ресурсу.

3. 3.Авторизація на основі контексту: в цьому методі доступ користувача до ресурсу визначається на основі контексту, такого як місцезнаходження користувача, час доби, IP-адреса, рівень доступ і т.д.

Деякі методи аутентифікації та авторизації можуть бути комбіновані, щоб забезпечити більш високий рівень безпеки. Наприклад, біометрична аутентифікація може бути поєднана з авторизацією на основі дозволів, щоб надати доступ до ресурсу тільки користувачам, які мають певний рівень дозволу та відповідають заданим біометричним характеристикам. Крім того, існують інші методи аутентифікації та авторизації, такі як аутентифікація на основі одноразових паролів, використання карточок доступу та багато інших. Кожен метод має свої переваги та недоліки, тому при виборі методу аутентифікації та авторизації слід враховувати конкретні потреби та вимоги проекту, а також рівень безпеки, який потрібно забезпечити. Крім методів аутентифікації та авторизації, існують інші заходи для забезпечення безпеки в системах, такі як шифрування, контроль доступу та моніторинг активності користувачів. Шифрування - це процес перетворення інформації у такий формат, який неможливо прочитати без спеціального ключа. Шифрування застосовується для захисту конфіденційної інформації, такої як паролі, номери банківських карток та інші особисті дані. Контроль доступу використовується для обмеження доступу до різних ресурсів системи на основі правил та рівня дозволів користувачів. Наприклад, адміністратор системи може надати доступ до певної частини системи лише окремим користувачам, які мають певні рівні дозволу [14].

Моніторинг активності користувачів дозволяє виявляти незвичайну активність та потенційні загрози безпеці в системі. Це може бути виконано шляхом ведення журналів подій, аналізу даних журналів та використання програмного забезпечення для виявлення підозрілих дій користувачів. Окрім того, важливо забезпечити безпеку під час розробки програмного забезпечення. Розробники повинні враховувати можливі загрози безпеці в процесі розробки та використовувати методи, що забезпечують безпеку, такі як перевірка вхідних даних, захист від вразливостей, кодування даних та інші. Також важливо забезпечити безпеку при зберіганні та обробці конфіденційної інформації. Наприклад, дані можна зберігати в захищеному середовищі з обмеженим доступом, а також використовувати шифрування даних та захист від несанкціонованого доступу. Крім того, важливо забезпечити безпеку в мережевому середовищі. Наприклад, можна використовувати захищені протоколи

зв'язку, такі як SSL / TLS, а також застосовувати методи захисту мережі, такі як брандмауери та системи виявлення вторгнень. Нарешті, важливо забезпечити безпеку в робочому середовищі. Наприклад, користувачі повинні забезпечити захист своїх пристроїв від несанкціонованого доступу, використовуючи паролі та інші методи захисту, а також повинні дотримуватися правил безпеки при використанні комп'ютера та Інтернету. Для забезпечення безпеки програмного забезпечення необхідно також регулярно оновлювати і підтримувати систему в актуальному стані.

Це означає встановлювати оновлення безпеки, виправляти помилки та вразливості, а також забезпечувати резервне копіювання і відновлення даних. Окрім того, для забезпечення безпеки важливо мати чітку політику доступу до системи.

Права доступу до даних повинні бути регулювані на основі принципу найменшого доступу, тобто користувачам повинно бути надано тільки ті права, які необхідні для виконання їхніх обов'язків. Також важливо забезпечити безпеку під час передачі даних через мережу. Наприклад, можна використовувати шифрування даних, підписи та сертифікати для перевірки автентичності даних, а також встановлювати захищені канали зв'язку. Крім того, важливо забезпечити безпеку на рівні користувача.

Наприклад, користувачі повинні використовувати складні паролі та двофакторну аутентифікацію для забезпечення захисту своїх облікових записів. Також важливо навчати користувачів правилам безпеки в Інтернеті та попереджувати їх про можливі загрози безпеці. Нарешті, важливо мати план надзвичайних ситуацій, що включає заходи для виявлення та відновлення системи в разі критичної атаки або випадку втрати даних. Для забезпечення безпеки програмного забезпечення важливо також враховувати потенційні загрози та вразливості. Наприклад, зловмисники можуть намагатися використати вразливості в програмному забезпеченні для отримання несанкціонованого доступу до даних, встановлення шкідливого програмного забезпечення або пошкодження системи. Тому важливо проводити аудит безпеки, щоб виявляти потенційні вразливості та відновлювати систему. Також важливо забезпечити захист від шкідливого програмного забезпечення, такого як віруси, черви та троянські програми. Для цього можна використовувати антивірусне програмне забезпечення та фаєрволи, які можуть блокувати шкідливий трафік та попереджати про можливі загрози. Також важливо використовувати захисні

механізми, такі як сенсори біометричних даних, які можуть виявляти спроби несанкціонованого доступу до системи, та захисні технології, які можуть запобігати зламу. Забезпечення безпеки також повинно включати контроль доступу до даних, зокрема забезпечення захисту конфіденційної інформації та забезпечення цілісності даних. Наприклад, можна використовувати криптографічні алгоритми для захисту даних від несанкціонованого доступу та надання доступу до даних тільки авторизованим користувачам. Окрім того, важливим елементом забезпечення безпеки є забезпечення захисту від внутрішніх загроз, які можуть виникнути через дії самого персоналу.

Для цього можна використовувати політики доступу до даних, контролювати дії користувачів та використовувати моніторинг системи для виявлення можливих порушень безпеки. Усі ці аспекти забезпечення безпеки програмного забезпечення важливі для забезпечення ефективного захисту від зловмисних атак та збереження конфіденційності, цілісності та доступності даних. Це вимагає постійного моніторингу та оновлення захисних механізмів та використання найсучасніших методів та технологій безпеки. Нарешті, важливо забезпечити постійний моніторинг системи для виявлення можливих атак та вразливостей та вжиття заходів для їх запобігання. Використання системи журналювання та моніторингу дозволяє виявляти та вирішувати проблеми з безпекою програмного забезпечення до того, як вони стануть серйозними загрозами для системи та даних користувачів.

3.2 Застосування шифрування для захисту даних

Шифрування – це процес перетворення даних у такий формат, що їх можна прочитати лише за наявності спеціального ключа. Застосування шифрування для захисту даних є важливим для забезпечення безпеки в Інтернеті та інших мережах передачі даних. Ось деякі з важливих причин, чому варто використовувати шифрування для захисту даних:

1. Захист від зламування: Шифрування може захистити дані від зламування та несанкціонованого доступу. Якщо дані будуть вкрадені або підібрані, їх не буде

можливо розшифрувати без ключа.

2. Конфіденційність: Шифрування дозволяє зберегти конфіденційність даних. Конфіденційні дані можуть бути, наприклад, фінансові дані, медичні записи, особисті повідомлення та інше.

3. Передача даних по мережі: Шифрування даних дозволяє передавати дані по мережі, не ризикуючи їх зламання та зловживанням. Для захисту передачі даних в Інтернеті використовуються різні протоколи шифрування, такі як SSL (Secure Sockets Layer) та TLS (Transport Layer Security).

4. Захист від вірусів та шпигунського програмного забезпечення: Шифрування може захистити від шкідливих програм, таких як віруси та шпигунське програмне забезпечення, які можуть перехоплювати дані, які ви вводите на комп'ютері.

5. Виконання регуляторних вимог: Деякі організації повинні дотримуватися регуляторних вимог з приводу захисту даних, таких як HIPAA, PCI DSS та GDPR. Використання шифрування може допомогти їм дотримуватися цих вимог.
Шифрування може бути використане в різних сферах, таких як:

1. Фінансовий сектор: Фінансові установи використовують шифрування для захисту фінансових даних, таких як банківські реквізити та транзакції, від зловживання та крадіжки.

2. Медична сфера: Шифрування використовується в медичній сфері для захисту конфіденційних медичних даних, таких як історії хвороб та лікарські рецепти. 3. Бізнес-сектор: Компанії використовують шифрування для захисту конфіденційної інформації, такої як фінансові дані, документи та плани продажів. 4. Комунальні послуги: Шифрування використовується в комунальних послугах, таких як електронні рахунки та показники лічильників, для захисту особистих даних клієнтів.

5. Телекомунікаційний сектор: Телекомунікаційні компанії використовують шифрування для захисту особистих даних клієнтів, таких як номери телефонів та інформація про виклики.

Додатково до перерахованих вище сфер, шифрування також використовується у багатьох інших галузях, таких як правоохоронні органи, наукові дослідження,

військова та урядова сфери, а також в Інтернеті. У світі, де кількість даних швидко зростає, захист цих даних є дедалі важливішим. Шифрування може допомогти захистити конфіденційну інформацію від несанкціонованого доступу, забезпечити безпеку віддалених транзакцій та комунікацій, а також зберегти цілісність даних під час їх передачі або зберігання. Однак, важливо зазначити, що жодна система шифрування не є непереборною. Відомі випадки порушення безпеки шифрування вказують на те, що необхідно регулярно підтримувати та оновлювати системи шифрування, щоб запобігти новим загрозам безпеці. Крім того, важливо забезпечити належний захист ключів шифрування, які використовуються для розшифрування зашифрованої інформації. Якщо ключі потраплять у руки зловмисників, це може призвести до зламу системи шифрування та отримання несанкціонованого доступу до конфіденційної інформації.

Одним з найважливіших аспектів шифрування є його використання для захисту особистих даних користувачів в Інтернеті. У сучасному світі більшість людей користується Інтернетом для здійснення різних операцій, таких як покупки в Інтернет-магазинах, банківські транзакції, відправка електронної пошти та багато іншого. Шифрування дозволяє захистити особисті дані користувачів від несанкціонованого доступу з боку зловмисників. Наприклад, при здійсненні онлайн покупки за допомогою захищеного протоколу шифрування, інформація про кредитну картку буде передана тільки продавцю, а не третім особам. Шифрування також допомагає захистити комунікації в Інтернеті. Захищений протокол шифрування може захистити конфіденційну інформацію, яку ви відправляєте або отримуєте через електронну пошту, чати або соціальні мережі. У сфері наукових досліджень шифрування використовується для захисту конфіденційної інформації та інтелектуальної власності. Дослідження з медицини та генетики містять конфіденційну інформацію про пацієнтів, яку необхідно захищати від несанкціонованого доступу. У військовій та урядовій сферах шифрування використовується для захисту конфіденційних даних, таких як плани військових операцій, інформація про національну безпеку та інші секретні документи. Крім захисту конфіденційної інформації, шифрування також може бути використано для забезпечення цілісності даних [15].

Це означає, що дані не можуть бути змінені або підроблені без знання ключа

шифрування. Шифрування даних є важливим елементом в сучасному світі і тісно пов’язане з безпекою і захистом інформації. Використання шифрування дозволяє забезпечити захист даних від несанкціонованого доступу, злому та крадіжки. Одним з основних методів шифрування є симетричне шифрування, де для шифрування та розшифрування даних використовується один і той же ключ. Цей метод є досить ефективним і швидким, проте його головним недоліком є необхідність обміну ключем між відправником та отримувачем. Це може бути небезпечно в разі перехоплення ключа третьою стороною. Іншим методом є асиметричне шифрування, яке використовує два ключі: публічний та приватний. Публічний ключ може бути доступний для будь-якої сторони, тоді як приватний ключ залишається тільки у власника. Цей метод більш безпечний, оскільки немає необхідності обмінюватися ключами між сторонами, але він може бути повільнішим та менш ефективним у порівнянні з симетричним шифруванням. Окрім того, шифрування є важливим елементом в інформаційній безпеці мереж і систем. За допомогою шифрування можна захистити передачу даних в Інтернеті, електронну пошту, телекомунікаційні мережі та інші засоби зв’язку від зломів та несанкціонованого доступу.

До переваг шифрування також можна віднести збереження цілісності даних, тобто забезпечення того, що дані не будуть змінені під час їх передачі чи зберігання. За допомогою цифрових підписів та хеш-функцій, що також є частинами криптографічних протоколів, можна забезпечити цілісність даних. Шифрування також використовують для забезпечення автентифікації користувачів та контролю доступу до захищених ресурсів. Наприклад, деякі веб-сайти вимагають від користувачів ввести логін та пароль для доступу до свого облікового запису. За допомогою шифрування, така інформація може бути захищена від несанкціонованого доступу та перехоплення. Крім того, шифрування також застосовують у фінансових операціях для забезпечення безпеки та недопущення фінансових шахрайств.

Наприклад, у банківській системі застосовуються різні криптографічні протоколи для захисту транзакцій та збереження конфіденційної інформації клієнтів. Однак, варто зазначити, що шифрування не є універсальним рішенням для всіх видів кіберзахисту та захисту даних. Воно може бути обмануте або скомпрометоване, якщо зловмисник отримає доступ до ключів шифрування, або якщо існують вразливості у

криптографічному протоколі. Тому важливо регулярно оновлювати та перевіряти системи шифрування, щоб забезпечити їх ефективність та безпеку. Отже, шифрування є надзвичайно важливим елементом безпеки даних та комунікацій. Використання шифрування може допомогти захистити дані від зловмисних атак, дотримуватися правил та законів щодо конфіденційності даних, захистити від шпигунства та крадіжки інтелектуальної власності та забезпечити безпеку в Інтернеті речей [16].

3.3 Методи контролю доступу до системи

Методи контролю доступу до системи Інтернету Речей (loT) можуть включати в себе наступні підходи:

1.Аутентифікація користувача: вимагається введення ідентифікатора користувача та пароля для доступу до системи loT. Це може бути реалізовано, наприклад, за допомогою мережевих протоколів таких як OAuth або OpenID.

2.Авторизація: після успішної аутентифікації, система може перевіряти дозволи користувача на доступ до певних ресурсів, які пов'язані зі системою loT. 3.Керування доступом: цей метод використовується для управління доступом до різних частин системи loT, які можуть бути розташовані в різних місцях. Це може бути досягнуто шляхом використання різноманітних механізмів, таких як протоколи маршрутизації та протоколи тунелювання.

4.Шифрування: захист даних, що передаються між системою loT та зовнішніми джерелами, може здійснюватися за допомогою шифрування. Це може бути здійснено за допомогою протоколів шифрування, таких як SSL або TLS.

5.Блокування: блокування може бути використане для заборони доступу до системи loT з певних місць чи від певних користувачів.

6.Мережеві заходи захисту: це може включати в себе різноманітні механізми, такі як фільтрація пакетів, детектори вторгнень, системи виявлення аномалій тощо. 7.Постійне оновлення програмного забезпечення: системи loT мають бути постійно оновлюванні та патчені для запобігання виявленим вразливостям. Це дозволяє зменшити ризик вторгнення через відомі вразливості.

8.Моніторинг системи: важливо постійно моніторити систему IoT на наявність незвичайної активності, аномального трафіку та інших відхилень. Це дозволяє своєчасно виявляти та реагувати на потенційні загрози.

9.Обмеження доступу до чутливих даних: у системі IoT важливо обмежити доступ до чутливих даних тільки для тих користувачів, які дійсно мають до них доступ. Це можна здійснити, наприклад, за допомогою системи ролей та прав доступу.

10. Використання захищених протоколів зв'язку: важливо використовувати захищені протоколи зв'язку, такі як HTTPS, MQTT з TLS, CoAP з DTLS тощо, для запобігання перехопленню та підробці даних.

11. Використання фізичних ідентифікаторів: для доступу до системи IoT можна використовувати фізичні ідентифікатори, такі як RFID-карти, що дозволяє підвищити безпеку системи та запобігти несанкціонованому доступу.

12. Резервне копіювання та відновлення даних: важливо регулярно створювати резервні копії даних та виконувати планові тестування процедур відновлення, що дозволяє забезпечити безпеку даних та зменшити ризик втрати даних.

13. Відстеження інцидентів: важливо вести журнал подій та реагувати на потенційні загрози швидко та ефективно. В разі виявлення несправностей або незвичайної активності в системі IoT, слід провести ретельне розслідування та вжити заходів для запобігання подібних інцидентів у майбутньому.

14. Використання захисту периметра: у системі IoT можна використовувати захист периметра для забезпечення безпеки даних та зменшення ризиків вторгнення. Це можна здійснити, наприклад, за допомогою брандмауера, що контролює вхідний та вихідний трафік.

15. Використання мультифакторної аутентифікації: для запобігання несанкціонованому доступу до системи IoT можна використовувати мультифакторну аутентифікацію. Наприклад, користувач може вводити свій логін та пароль, а потім підтверджувати свою ідентичність за допомогою SMS-повідомлення або додатку для автентифікації.

16. Аудит безпеки системи: важливо проводити аудит безпеки системи IoT для виявлення можливих вразливостей та відхилень від встановлених правил безпеки.

Аудит безпеки можна проводити самостійно або залучати фахівців з цієї області.

17. Постійне оновлення програмного забезпечення: для запобігання можливих вразливостей та підвищення рівня безпеки системи IoT, необхідно постійно оновлювати програмне забезпечення. Виробники системи IoT повинні забезпечувати регулярні оновлення програмного забезпечення та запобігати використанню застарілих версій програм.

18. Навчання користувачів: важливо проводити навчання користувачів щодо правил безпеки та можливих загроз у використанні системи IoT. Навчання може включати в себе інструктаж щодо створення складних паролів, обізнаність з правилами доступу до системи та засобами забезпечення безпеки.

19. Моніторинг активності: необхідно забезпечувати постійний моніторинг активності в системі IoT. Це допоможе вчасно виявляти неправильну поведінку та можливі загрози безпеці. Наприклад, система моніторингу може виявляти спроби несанкціонованого доступу або змін в налаштуваннях системи.

20. Захист даних: система IoT містить велику кількість даних, що потребують захисту. Для захисту цих даних можна використовувати криптографічні протоколи та шифрування. Дані повинні бути захищені від несанкціонованого доступу, втрати, викрадення та знищення.

21. Відповідальність виробника: виробник системи IoT повинен нести відповідальність за безпеку своїх пристроїв та програмного забезпечення. Він повинен забезпечувати безпеку від зовнішніх загроз та оновлювати програмне забезпечення, щоб запобігти вразливостям.

22. Захист від DDoS-атак: система IoT може бути піддається DDoS-атакам, коли злочинці можуть заволодіти пристроями та використовувати їх для створення ботнетів. Для запобігання DDoS-атакам необхідно використовувати захист від DDoS атак, який дозволяє виявляти та блокувати небезпечний трафік.

23. Фізичний захист: система IoT повинна бути захищена від несанкціонованого фізичного доступу до пристроїв. Наприклад, пристрої повинні бути установлені у захищеному приміщенні та бути забезпечені міцними замками.

24. Аудит безпеки: необхідно проводити регулярний аудит безпеки системи IoT

з метою виявлення та усунення можливих вразливостей. Це дозволить забезпечити постійний контроль за системою та зменшити ризик зламу.

25. Захист мережі: необхідно забезпечити захист мережі, яка використовується для підключення до системи IoT. Наприклад, можна використовувати мережеві елементи, які забезпечують захист мережі від атак.

26. Захист від внутрішніх загроз: система IoT може бути піддається внутрішнім загрозам, коли злочинці можуть зламати систему зсередини. Для запобігання цим загрозам необхідно використовувати методи аутентифікації та авторизації користувачів, а також захист від вразливостей.

27. Навчання персоналу: персонал, який має доступ до системи IoT, повинен бути навчений методам безпеки та розуміти потенційні загрози безпеці. Це допоможе зменшити ризик виникнення проблем через неправильну поведінку персоналу.

Взагалі, система IoT потребує комплексного підходу до захисту, який включає не тільки технічні, але й організаційні та людські аспекти. Забезпечення безпеки в системі IoT є надзвичайно важливим для забезпечення функціональності та надійності системи [17].

3.4 Системи моніторингу та логування

Системи моніторингу та логування - це інструменти, які дозволяють збирати, аналізувати та зберігати дані про різні події, які відбуваються в комп'ютерних системах або мережах. Ці інструменти можуть бути використані для виявлення проблем, аналізу причин їх виникнення, а також для попередження майбутніх проблем. Системи моніторингу дозволяють відслідковувати різні параметри системи, такі як використання ресурсів, навантаження на процесор, пам'ять та диск, стан мережі, доступність сервісів та інші. За допомогою цих інструментів можна відстежувати роботу системи в реальному часі, що дозволяє оперативно виявляти проблеми та реагувати на них.

Системи логування дозволяють збирати та зберігати інформацію про різні події, які відбуваються в системі або мережі, такі як входи користувачів, запити до серверів,

помилки та винятки, інформацію про стан системи та інше. Ці інструменти дозволяють зберігати детальну інформацію про стан системи протягом тривалого часу, що може бути корисним для аналізу причин виникнення проблем. Системи моніторингу та логування використовуються в багатьох галузях, включаючи інформаційну технологію, фінанси, транспорт, промисловість та багато інших. Ці інструменти дозволяють збирати та аналізувати великі обсяги даних, що дозволяє знаходити та вирішувати проблеми в реальному часі. Один з основних викликів при використанні систем моніторингу та логування - це обробка великої кількості даних. Для забезпечення ефективної обробки даних, використовуються різні технології, такі як бази даних, потокова обробка даних та інші. Ще одним викликом є забезпечення безпеки зібраних даних. У багатьох випадках, дані, що збираються системами моніторингу та логування, можуть містити конфіденційну інформацію, таку як імена користувачів, паролі, адреси електронної пошти та інше. Тому важливо забезпечувати захист даних від несанкціонованого доступу та зловживання. Системи моніторингу та логування можуть бути використані для відстежування виконання SLA (Service Level Agreement), що дозволяє контролювати рівень обслуговування, який надається клієнтам та користувачам [18].

Це дозволяє забезпечувати високу якість обслуговування та задоволення потреб клієнтів. Одним з типів систем моніторингу є системи моніторингу мережі, які використовуються для відстежування стану мережевих пристроїв та з'єднань між ними. Ці системи можуть допомогти виявити проблеми, такі як перевантаження мережі, некоректна конфігурація або проблеми зі з'єднанням. Іншим типом систем моніторингу є системи моніторингу додатків, які використовуються для відстежування стану програмного забезпечення. Ці системи можуть допомогти виявити проблеми, такі як помилки програми, несправності, помилки в базі даних та інші. Системи логування дозволяють збирати та зберігати записи про події, які сталися в системі, такі як запити користувачів, помилки програм та інші. Ці записи можуть бути використані для аналізу причин проблем та їх вирішення.

Одним з ключових елементів систем моніторингу та логування є метрики. Метрики - це числові значення, які вказують на стан певного аспекту системи, такого як навантаження процесора, використання пам'яті, кількість запитів до бази даних та

інші. Вимірювання метрик дозволяє отримувати інформацію про стан системи, а їх аналіз допомагає виявляти проблеми та покращувати її ефективність. Одним з популярних інструментів моніторингу є Prometheus. Він дозволяє збирати метрики з різноманітних джерел, а також використовувати їх для створення графіків та сповіщень про проблеми. Іншим популярним інструментом є Grafana, який дозволяє візуалізувати зібрані метрики та створювати графіки та панелі керування. Системи логування зазвичай використовуються для збору та збереження записів про події в системі. Записи можуть бути збережені у різних форматах, таких як текстові файли, бази даних та інші. Для аналізу записів зазвичай використовуються інструменти, такі як Elasticsearch та Kibana, які дозволяють швидко знаходити та аналізувати записи [19].

Додатково до метрик і логів, системи моніторингу та логування можуть використовувати інші інструменти та технології, такі як агенти моніторингу, інструменти трасування запитів (tracing), та інші. Агенти моніторингу - це програмні компоненти, які запущені на кожній машині або сервісі в системі та збирають метрики та логи з цих машин або сервісів. Наприклад, у випадку з Prometheus, агент моніторингу - це програма, яка запущена на кожній машині, та яка збирає метрики з цієї машини та передає їх до центрального сервера. Інструменти трасування запитів дозволяють відстежувати шлях запиту в системі, тобто які компоненти системи обробляли запит, який час вони на це витрачали та чи було з ними проблем. Трасування допомагає зрозуміти, як система працює та де можуть бути проблеми.

Також системи моніторингу та логування можуть використовувати сповіщення та інші механізми оповіщення про проблеми в системі. Наприклад, Prometheus може відправляти сповіщення на електронну пошту або на платформи спілкування (Slack, Telegram, Viber та інші) про перевищення заданих лімітів метрик. Додатково до того, що я вже згадував, системи моніторингу та логування можуть бути використані для забезпечення безпеки інформаційної системи. Наприклад, збирання та аналіз логів може допомогти виявити зловмисну діяльність або несанкціонований доступ до системи. Для забезпечення безпеки, системи моніторингу та логування можуть використовувати такі інструменти, як системи виявлення вторгнень (Intrusion Detection Systems, IDS), системи виявлення компрометації (Compromise Detection

Systems, CDS) та інші. Ці інструменти відслідковують підозрілі активності, такі як спроби несанкціонованого доступу до системи, спроби виконання вразливостей в програмному забезпеченні та інші, та надсилають повідомлення про ці активності до адміністратора системи.

Ще одним аспектом систем моніторингу та логування є можливість аналізу метрик та логів для виявлення проблем з продуктивністю системи. Наприклад, аналіз метрик завантаження процесора та пам'яті може допомогти виявити проблеми з продуктивністю системи та знайти способи їх вирішення. Також аналіз логів може допомогти виявити проблеми з продуктивністю, такі як запити до бази даних, які займають надмірний час. Крім того, системи моніторингу та логування можуть бути використані для відновлення системи після аварії. Якщо система перестала працювати, адміністратор може використати логи, щоб знайти причину проблеми та відновити роботу системи. Нарешті, системи моніторингу та логування можуть допомогти виявити недоліки та можливість покращення системи. Наприклад, аналіз логів може допомогти виявити, які функції системи використовуються найменше та які можуть бути видалені або оптимізовані, що допоможе покращити продуктивність та знизити витрати на підтримку системи. Додатково до вищезгаданих переваг, системи моніторингу та логування можуть бути корисні в багатьох інших випадках. Наприклад, системи моніторингу можуть використовуватися для контролю навантаження на сервери та мережі.

Адміністратор може використовувати систему моніторингу, щоб визначити, чи працюють сервери на максимальній потужності, чи можна додати ще серверів для розширення мережі, чи необхідно змінювати конфігурацію серверів для забезпечення оптимальної продуктивності. Логи можуть також допомогти в розслідуванні інцидентів безпеки. Наприклад, якщо система була скомпрометована, логи можуть містити корисну інформацію про те, які дії були виконані злочинцями, і з яких IP адрес були здійснені спроби входу в систему. Ця інформація може бути використана для ідентифікації злочинців та уникнення подібних інцидентів у майбутньому. Крім того, системи моніторингу та логування можуть допомогти забезпечити відповідність регуляторним вимогам та стандартам. Наприклад, GDPR вимагає зберігання логів, щоб забезпечити захист персональних даних. Інші стандарти, такі як SOX, можуть

вимагати зберігання логів для забезпечення фінансової звітності та аудиту. У підсумку, системи моніторингу та логування є важливими компонентами будь-якої інформаційної системи.

Вони забезпечують стеження за роботою системи, виявлення проблем, допомогу їх усуненні, збереження даних для виконання вимог законодавства та стандартів безпеки, а також допомогу в плануванні та прийнятті рішень. Загалом, системи моніторингу та логування є невід'ємною частиною будь-якої інформаційної системи та можуть бути використані для забезпечення безпеки, відповідності регуляторним вимогам та підвищення продуктивності. Оскільки інформаційні системи стають все більш складними, важливість систем моніторингу та логування тільки зростає [20].

3.5 Фізичний захист складових системи IoT

Фізичний захист є одним з ключових аспектів забезпечення безпеки систем Інтернету Речей (IoT). Для того, щоб забезпечити високий рівень захисту, необхідно виконувати різноманітні заходи з фізичного захисту складових системи IoT. Один з найбільш ефективних методів захисту - це застосування захисних корпусів інтернет пристроїв. Вони можуть бути виготовлені з міцних матеріалів, які здатні витримати механічні пошкодження та інші фізичні впливи. Також можуть бути використані захисні екрани або інші види захисних конструкцій, які зменшують ризик зламу пристрою або злочинного використання його функцій. Для захисту системи IoT також можна використовувати захисні мережеві системи, такі як мережеві фаєрволи та інтра-мережеві детектори вторгнень (IDS). Ці системи можуть розпізнавати та блокувати зловмисну активність в мережі, що може бути причиною несанкціонованого доступу до системи IoT.

Також важливо забезпечити безпеку фізичного простору, в якому знаходиться система IoT. Це можна зробити шляхом застосування контролю доступу до приміщення, використання систем відеоспостереження та інших методів фізичного захисту. Окрім того, необхідно забезпечити фізичний захист при зберіганні даних, зокрема захист від несанкціонованого доступу до зберігаються на пристрої дані. Окрім

того, важливо забезпечити правильну утилізацію відпрацьованих пристроїв IoT, щоб забезпечити відсутність витоку конфіденційної інформації та зменшення негативного впливу на довкілля. Загалом, фізичний захист є важливою складовою безпеки систем Інтернету Речей, який потребує постійного уваги та вдосконалення. Застосування захисних корпусів, захисних екранів, мережевих фаєрволів, інтра мережевих детекторів вторгнень та інших заходів можуть значно зменшити ризик зламу та несанкціонованого доступу до системи IoT. Додатково до вищезгаданих заходів фізичного захисту, до складу яких можуть входити захисні корпуси, захисні екрани, мережеві фаєрволи, інтра-мережеві детектори вторгнень та інші, існують також інші методи захисту систем Інтернету Речей.

Один з таких методів - це використання бездротових технологій передачі даних, таких як NFC (Near Field Communication) або BlueTooth. NFC є технологією короткого діапазону зв'язку, яка забезпечує зв'язок між двома пристроями на відстані до 10 см, тоді як BlueTooth - технологія передачі даних на більшій відстані, до 100 метрів, залежно від версії протоколу BlueTooth. Використання бездротових технологій дозволяє знизити ризик зламу системи через фізичний доступ до пристроїв IoT. Іншим методом фізичного захисту є використання біометричної ідентифікації, такої як сканування відбитків пальців або розпізнавання обличчя. Це може додатково забезпечити захист від несанкціонованого доступу до системи, оскільки біометрична ідентифікація базується на унікальних фізичних характеристиках кожної особи. Крім того, важливо враховувати і фізичні аспекти захисту, такі як безпека проти пожежі та забезпечення безперебійного живлення пристроїв IoT. Встановлення датчиків пожежі та систем аварійного живлення можуть допомогти уникнути негативних наслідків в разі пожежі або збою в електромережі. Крім того, важливим аспектом фізичного захисту систем IoT є захист від фізичної крадіжки та злому. Для цього можуть використовуватися різноманітні методи, такі як встановлення системи відеоспостереження або використання GPS-трекінгу для відстеження розташування пристроїв. Крім того, важливим аспектом фізичного захисту є безпека під час транспортування пристроїв IoT. При транспортуванні необхідно забезпечувати безпеку та інтегрітет пристроїв, а також захист від небезпек, пов'язаних з ризиками втрати даних, фізичних пошкоджень та втрати пристроїв. Окрім цього, можуть

використовуватися різноманітні методи захисту від електромагнітних перешкод.

Наприклад, можуть бути встановлені екрануючі матеріали, які зменшують вплив електромагнітних полів, або між модульні екрани, які забезпечують екранування сигналів від електромагнітних перешкод. Отже, фізичний захист систем IoT є важливим елементом їх безпеки. Для забезпечення фізичної безпеки необхідно враховувати різноманітні аспекти, такі як безпека проти фізичних пошкоджень та крадіжки, захист від електромагнітних перешкод, захист під час транспортування та захист від несанкціонованого доступу. Ще одним важливим аспектом фізичного захисту систем IoT є захист від небезпек, пов'язаних зі стихійними лихами та надзвичайними ситуаціями. Наприклад, системи IoT можуть бути пошкоджені під час повеней, землетрусів, пожеж, ураганів та інших небезпечних ситуацій. Для забезпечення захисту від цих небезпек, можуть використовуватися різні методи. Наприклад, системи IoT можуть бути встановлені на високому рівні, щоб забезпечити захист від повеней, або забезпечити бекап даних в захищеному місці, щоб запобігти втраті даних у разі пожежі або іншої надзвичайної ситуації. Також важливим є захист від несанкціонованого доступу до систем IoT через фізичний доступ. Наприклад, можуть використовуватися різноманітні методи контролю доступу, такі як картки доступу, системи розпізнавання обличчя або системи біометричної ідентифікації. Крім того, важливо забезпечувати фізичну безпеку систем IoT від зловмисних атак, таких як фізичний доступ до пристроїв IoT для зміни їх налаштувань або використання для злочинних дій [21].

Для цього можуть використовуватися методи криптографічного захисту, такі як шифрування даних та використання сертифікатів безпеки. Один із способів забезпечення фізичного захисту компонентів системи IoT - це використання механізмів обмеження доступу до пристроїв і техніки. Такі механізми можуть включати фізичні бар'єри, наприклад, замки і пристрої для контролю доступу, такі як картки з доступом або біометричні сканери, або системи безпеки, які дозволяють відслідковувати доступ до об'єктів у режимі реального часу. Ще одним важливим аспектом фізичного захисту є захист від небезпек, що походять з навколишнього середовища, таких як пожежа, повінь, землетруси тощо. Виробники систем IoT повинні приділяти увагу проектуванню інженерних мереж та компонентів системи,

щоб забезпечити максимальну стійкість до таких небезпек. Також важливо враховувати віддалені місця розташування складових системи IoT та можливість їх підключення до мережі Інтернет. Для захисту від несанкціонованого доступу до таких вузлів системи використовуються методи аутентифікації та авторизації, а також захисту мережевого трафіку шляхом використання шифрування і тунелювання.

Окрім цього, компоненти системи IoT, які знаходяться у віддалених місцях з високим рівнем критичності, можуть бути забезпечені додатковими фізичними засобами захисту, такими як відеоспостереження або безпілотні літальні апарати (дрони), які здатні виконувати моніторинг і забезпечувати швидку відповідь на небезпеки, які виникають у віддалених місцях. Наступним аспектом, який може забезпечити фізичний захист системи IoT, є контроль доступу. Це може включати в себе фізичний доступ до пристроїв IoT, а також доступ до даних, які зберігаються на них. Щоб забезпечити контроль доступу до пристроїв IoT, можна використовувати такі техніки, як ідентифікація та аутентифікація користувачів. Для ідентифікації користувачів можна використовувати унікальні ідентифікатори, такі як імена користувачів або електронні адреси.

Для аутентифікації користувачів можна використовувати паролі або інші методи аутентифікації, такі як біометричні дані. Крім того, важливо забезпечити захист даних, які зберігаються на пристроях IoT. Для цього можна використовувати методи шифрування даних, щоб забезпечити їх конфіденційність. Також можна використовувати методи цифрового підпису, щоб забезпечити цілісність даних та перевірити їх автентичність. Окрім цього, важливо забезпечити фізичний захист пристроїв IoT від зовнішніх пошкоджень та небезпечних впливів. Для цього можна використовувати захисні кожухи, які захищають пристрої від впливу погодних умов або механічних пошкоджень. Також важливо забезпечити електричну безпеку та захист від перенапруги. Отже, фізичний захист систем IoT є важливим елементом їх безпеки.

Для забезпечення фізичної безпеки необхідно враховувати різноманітні аспекти, такі як безпека проти фізичних пошкоджень та крадіжки, захист від електромагнітних перешкод, захист під час транспортування та захист від несанкціонованого доступу [22].

РОЗДІЛ 4
РОЗРОБКА ДОДАТКУ НА БАЗІ BLUETOOTH ТА ВИСНОВКИ

4.1 Розробка додатку, опис принципу та роботи

Для початку почнемо з схеми роботи та обговорення принципу дії зміни BlueTooth адреси. BlueTooth Internet of Things використовує BlueTooth-технологію для підключення пристроїв IoT до мережі Інтернет. Зміна адреси BlueTooth є однією з рекомендованих заходів для підвищення безпеки системи. Ось кілька причин, чому це важливо:

1. Анонімність: Зміна адреси BlueTooth дозволяє забезпечити анонімність пристрою, зменшуючи ризик відстеження та ідентифікації пристрою. Це особливо важливо для пристроїв IoT, оскільки вони можуть містити цінну інформацію про власників або мережу.

2. Захист від атак: Зміна адреси BlueTooth ускладнює можливість атак зламу або перехоплення з'єднання. Зловмисники, які намагаються зламати з'єднання або здійснити недоброзичливу дію, можуть виявити пристрій за його адресою BlueTooth. Зміна адреси знижує ймовірність таких атак.

3. Захист приватності: Зміна адреси BlueTooth допомагає зберегти приватні дані пристрою. Якщо пристрій постійно використовує одну адресу, це може зробити його вразливим до прослуховування або ідентифікації його власника. Зміна адреси дозволяє зберегти конфіденційність користувача та ускладнює можливість визначення його особистості.

4. Виключення заборонених пристроїв: Зміна адреси BlueTooth також дозволяє виключати заборонені пристрої з мережі. Якщо виявлено пристрій з небажаними діями або потенційно небезпечною поведінкою, можна заблокувати його адресу BlueTooth, щоб запобігти подальшому підключенню до мережі.

Зміна адреси BlueTooth є однією зі стратегій, які можна використовувати для підвищення безпеки системи BlueTooth IoT. Однак слід враховувати, що це лише

один аспект безпеки, і інші заходи, такі як використання шифрування та аутентифікації, також мають бути впроваджені для забезпечення повноцінного захисту системи [23].

Схема роботи додатків рисунок 4.1.

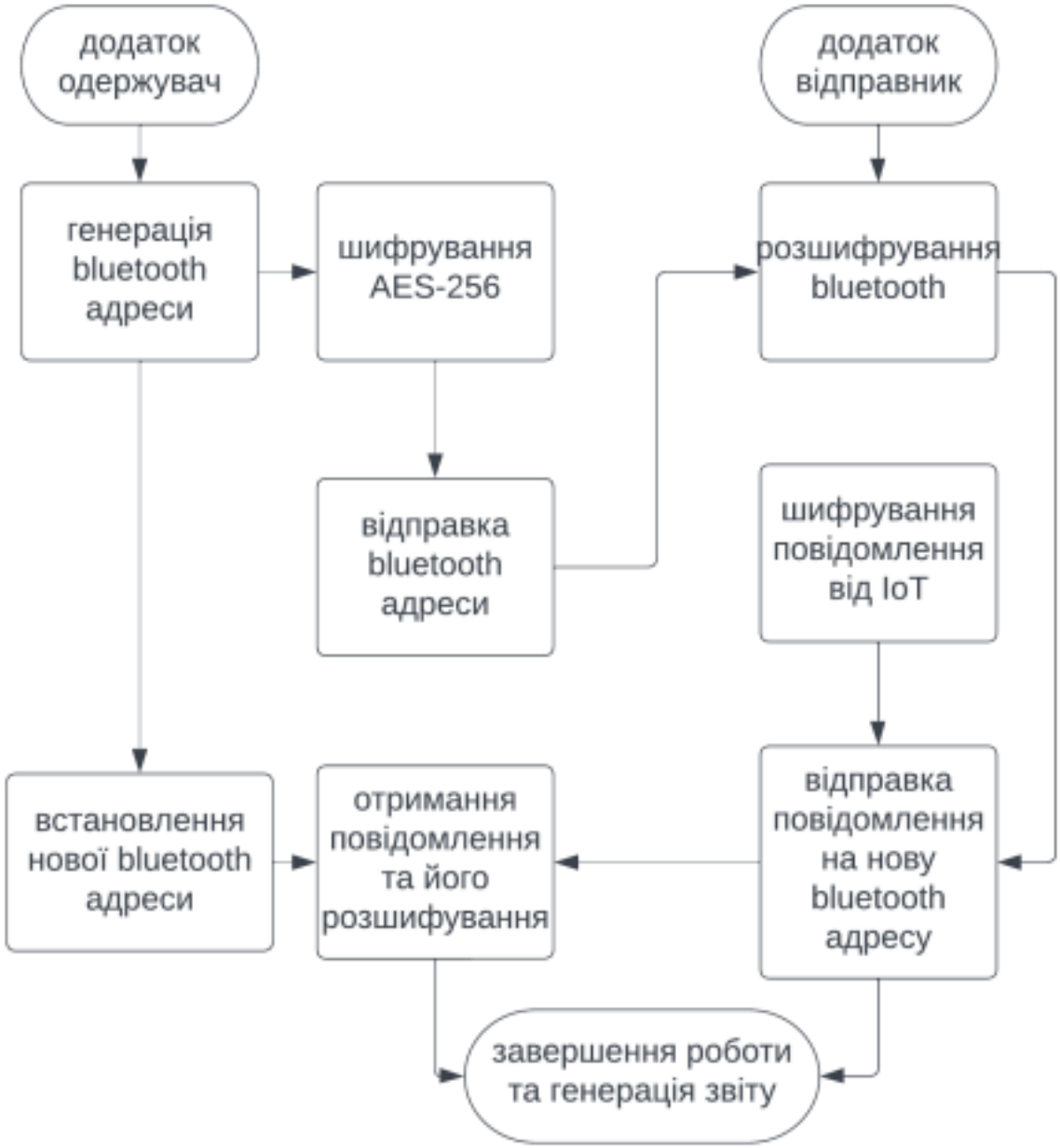

Рисунок 4.1 – Зображена спрощена робота додатків які працюють над зміною BlueTooth адреси та підвищенням безпеки

Почнемо з першого додатку, а саме з блоку генерації випадкової BlueTooth адреси яка нам знадобиться для заміни статичної, рисунок 4.2.

```
static string GenerateBluetoothAddress()
{
    // Генеруємо випадковий Bluetooth адресу
    var random = new Random();
    var addressBytes = new byte[6];
    random.NextBytes(addressBytes);
    addressBytes[0] |= 0x01; // Встановлюємо локальний біт (LSB) 1

    // Перетворюємо адресу на рядок у форматі "XX:XX:XX:XX:XX:XX"
    var addressBuilder = new StringBuilder();
    foreach (var b in addressBytes)
    {
        addressBuilder.AppendFormat("{0:X2}:", b);
    }
    addressBuilder.Length -= 1; // Видаляємо останній двокрапку
    return addressBuilder.ToString();
}
```

Рисунок 4.2 –Генерація адреси

Для генерації BlueTooth адреси встановлюємо 6 локальних (LBS) бітів та генеруємо через random адресу. Потім перетворюємо все в рядок та видаляємо останню двокрапку яка з’являється через особливості генерації.

Далі код виконує операцію шифрування BlueTooth адреси за допомогою заданого ключа шифрування рисунок 4.3. Основна функція EncryptMessage отримує повідомлення (у нашому випадку – BlueTooth адресу) та ключ шифрування і використовує алгоритм шифрування AES для шифрування даних рисунок 4.4.

```
// Статичний ключ для шифрування (256 біт)
byte[] encryptionKey = new byte[] {
    0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF,
    0xFE, 0xDC, 0xBA, 0x98, 0x76, 0x54, 0x32, 0x10,
    0x31, 0x2C, 0x5F, 0x7E, 0x12, 0x4A, 0x6D, 0x3F,
    0x8B, 0x9D, 0xC2, 0x3E, 0x5A, 0x17, 0x76, 0xE9
                                        };
```

Рисунок 4.3 – Статичний ключ шифрування

```
static byte[] EncryptMessage(string message, byte[] key)
{
    using (var aes = Aes.Create())
    {
        aes.Key = key;
        aes.GenerateIV();
        byte[] encryptedData;

        using (var encryptor = aes.CreateEncryptor())
        {
            using (var memoryStream = new MemoryStream())
            {
                using (var cryptoStream = new CryptoStream(memoryStream, encryptor, CryptoStreamMode.Write))
                {
                    byte[] messageBytes = Encoding.UTF8.GetBytes(message);
                    cryptoStream.Write(messageBytes, 0, messageBytes.Length);
                }
                encryptedData = memoryStream.ToArray();
            }
        }

        byte[] encryptedMessage = new byte[aes.IV.Length + encryptedData.Length];
        Array.Copy(aes.IV, encryptedMessage, aes.IV.Length);
        Array.Copy(encryptedData, 0, encryptedMessage, aes.IV.Length, encryptedData.Length);

        return encryptedMessage;
    }
}
```

Рисунок 4.4 – Функція EncryptMessage

У функції EncryptMessage створюється екземпляр AES з заданим ключем та генерується випадковий вектор ініціалізації (IV). Використовуючи об’єкт encryptor, отримані дані шифруються. Зашифровані дані записуються в MemoryStream, використовуючи об’єкт cryptoStream, який виконує шифрування під час записування даних. Після завершення шифрування зашифровані дані перетворюються в масив байт. В конструкторі encryptedMessage створюється масив, який складається з IV та зашифрованих даних. Цей масив байт повертається як результат шифрування. Статичний ключ шифрування задається у вигляді масиву байт. Викликається функція EncryptMessage, яка шифрує BlueTooth адресу. Зашифрована адреса може бути передана до іншої програми або використана для подальшої передачі по BlueTooth.

У функції Main спочатку генерується випадкова BlueTooth адреса за допомогою функції GenerateBlueToothAddress. Потім користувач вводить адресу BlueTooth отримувача. А далі додаток отримує повідомлення IoT назад та розшифровує його рисунок 4.5.

```
{
    // Генеруємо Bluetooth адресу
    string bluetoothAddress = GenerateBluetoothAddress();
    Console.WriteLine("Generated Bluetooth Address: " + bluetoothAddress);

    // Отримуємо Bluetooth адресу отримувача від користувача
    Console.Write("Enter the recipient's Bluetooth Address: ");
    string recipientAddress = Console.ReadLine();

    // Статичний ключ для шифрування (256 біт)
    byte[] encryptionKey = new byte[] {
     0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF,
     0xFE, 0xDC, 0xBA, 0x98, 0x76, 0x54, 0x32, 0x10,
     0x31, 0x2C, 0x5F, 0x7E, 0x12, 0x4A, 0x6D, 0x3F,
     0x8B, 0x9D, 0xC2, 0x3E, 0x5A, 0x17, 0x76, 0xE9
                                        };

    // Шифруємо Bluetooth адресу
    byte[] encryptedAddress = EncryptMessage(bluetoothAddress, encryptionKey);

    // Отримуємо повідомлення від другої програми
    byte[] receivedMessage = encryptedAddress;

    // Розшифровуємо повідомлення
    string decryptedMessage = DecryptMessage(receivedMessage, encryptionKey);
```

Рисунок 4.5 – Зображена частина коду з роботою над повідомленням та BlueTooth адресою

Розшифрування повідомлення відбудеться окремо у функції DecryptMessage рисунок 4.6.

```
static string DecryptMessage(byte[] encryptedMessage, byte[] key)
{
    using (var aes = Aes.Create())
    {
        aes.Key = key;

        byte[] iv = new byte[aes.BlockSize / 8];
        byte[] cipherText = new byte[encryptedMessage.Length - iv.Length];

        Array.Copy(encryptedMessage, iv, iv.Length);
        Array.Copy(encryptedMessage, iv.Length, cipherText, 0, cipherText.Length);

        aes.IV = iv;

        using (var decryptor = aes.CreateDecryptor())
        {
            using (var memoryStream = new MemoryStream(cipherText))
            {
                using (var cryptoStream = new CryptoStream(memoryStream, decryptor, CryptoStreamMode.Read))
                {
                    using (var reader = new StreamReader(cryptoStream))
                    {
                        return reader.ReadToEnd();
                    }
                }
```

Рисунок 4.6 – Функція DecryptMessage

Вона виконує розшифрування зашифрованого повідомлення з використанням заданого ключа шифрування. У функції `DecryptMessage` передаються два параметри: `encryptedMessage` (зашифроване повідомлення у вигляді масиву байт) та `key` (ключ шифрування у вигляді масиву байт).

Функція спочатку визначає розмір блоку ініціалізації (IV) та зашифрованих даних. Розмір IV становить 16 байт, тому функція створює масив `iv` розміром 16 байтів, а масив `encryptedData` ініціалізується з решти байтів `encryptedMessage`.

Потім функція створює екземпляр AES і встановлює ключ шифрування (`aes.Key`) та IV (`aes.IV`).

Використовуючи `aes.CreateDecryptor()`, створюється об’єкт `decryptor`, який виконує розшифрування даних. Далі функція створює `MemoryStream` (`memoryStream`), куди будуть записуватись розшифровані дані. Цей `MemoryStream` передається в `CryptoStream` (`cryptoStream`), який використовує об’єкт `decryptor` для розшифрування даних під час записування.

Функція записує `encryptedData` у `cryptoStream`, викликаючи метод `Write`, який записує дані у `memoryStream`. Після завершення розшифрування дані `memoryStream` перетворюються в масив байт за допомогою `ToArray()` і повертаються як результат функції `DecryptMessage` [24].

Отримані розшифровані дані можуть бути далі оброблені або використані за потреби.

У завершення зберігається звіт у вигляді .txt файла рисунок 4.7.

```
// Зберігаємо звіт в текстовий файл
string report = "Encryption Key: " + BitConverter.ToString(encryptionKey).Replace("-", "") + Environment.NewLine;
report += "Status: Successful" + Environment.NewLine;
report += "Time: " + DateTime.Now.ToString() + Environment.NewLine;
report += "Original Bluetooth Address: " + bluetoothAddress + Environment.NewLine;
report += "Encrypted Bluetooth Address: " + BitConverter.ToString(encryptedAddress).Replace("-", "") + Environment.NewLine;
report += "Recipient Bluetooth Address: " + recipientAddress + Environment.NewLine;
report += "Received Message: " + BitConverter.ToString(receivedMessage).Replace("-", "") + Environment.NewLine;
report += "Decrypted Message: " + decryptedMessage;

string desktopPath = Environment.GetFolderPath(Environment.SpecialFolder.Desktop);
string reportFilePath = Path.Combine(desktopPath, "encryption_report.txt");
File.WriteAllText(reportFilePath, report);

Console.WriteLine("Report saved to: " + reportFilePath);
```

Рисунок 4.7 –Генерація звіту

Бачимо данні звіту та шлях збереження, а ще бачимо у консолі куди саме було збережено звіт.

Тепер можемо розглянути додаток зі сторони IoT. Спочатку ми побачимо таку ж саму функцію розшифрування DecryptMessage, як у першому додатку рисунок 4.8.

```
static byte[] DecryptMessage(byte[] encryptedMessage, byte[] key)
{
    byte[] iv = new byte[16];
    byte[] encryptedData = new byte[encryptedMessage.Length - iv.Length];

    Array.Copy(encryptedMessage, iv, iv.Length);
    Array.Copy(encryptedMessage, iv.Length, encryptedData, 0, encryptedData.Length);

    using (var aes = Aes.Create())
    {
        aes.Key = key;
        aes.IV = iv;

        using (var descriptor = aes.CreateDecryptor())
        {
            using (var memoryStream = new MemoryStream())
            {
                using (var cryptoStream = new CryptoStream(memoryStream, descriptor, CryptoStreamMode.Write))
                {
                    cryptoStream.Write(encryptedData, 0, encryptedData.Length);
                }
                return memoryStream.ToArray();
            }
        }
    }
}
```

Рисунок 4.8 – Функція DecryptMessage

Ця функція виконує розшифрування зашифрованого повідомлення отримане по BlueTooth.

Потім наступна частина коду виконує шифрування повідомлення з використанням заданого ключа шифрування та відправки на отриману та розшифровану адресу (сама адреса IoT є статичною) рисунок 4.9.

```
static void Main()
{
    // Встановлюємо адресу Bluetooth-отримувача
    string recipientAddress = "D6:37:F2:8D:53:C1";

    // Відправка зашифрованого повідомлення (по Bluetooth)
    byte[] messageBytes = Encoding.UTF8.GetBytes("the connection is hidden");

    // Статичний ключ для шифрування (256 біт)
    byte[] decryptionKey = new byte[] {
     0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF,
     0xFE, 0xDC, 0xBA, 0x98, 0x76, 0x54, 0x32, 0x10,
     0x31, 0x2C, 0x5F, 0x7E, 0x12, 0x4A, 0x6D, 0x3F,
     0x88, 0x9D, 0xC2, 0x3E, 0x6A, 0x17, 0x76, 0xE9
                                        };

    // Шифруємо повідомлення
    byte[] encryptedMessage;
    using (var aes = Aes.Create())
    {
        aes.Key = decryptionKey;
        aes.GenerateIV();

        using (var encryptor = aes.CreateEncryptor())
        {
            using (var memoryStream = new MemoryStream())
            {
                memoryStream.Write(aes.IV, 0, aes.IV.Length);
                using (var cryptoStream = new CryptoStream(memoryStream, encryptor, CryptoStreamMode.Write))
                {
                    cryptoStream.Write(messageBytes, 0, messageBytes.Length);
                }
                encryptedMessage = memoryStream.ToArray();
            }
        }
    }
```

Рисунок 4.9 – У функції `Main` спочатку встановлюється адреса BlueTooth отримувача`recipientAddress`

Наступним кроком є перетворення текстового повідомлення `»the connection is hidden»` в масив байтів `messageBytes` за допомогою `Encoding.UTF8.GetBytes()`. Також визначається статичний ключ для шифрування `decryptionKey`, який представлений у вигляді масиву байтів.

Далі починається процес шифрування повідомлення. Створюється екземпляр AES (`aes`) за допомогою `Aes.Create()`. Задається ключ шифрування (`aes.Key`) та генерується випадковий вектор ініціалізації (IV) (`aes.GenerateIV()`).За допомогою `aes.CreateEncryptor()` створюється об’єкт `encryptor`, який виконує шифрування даних.

Далі створюється `MemoryStream` (`memoryStream`), який використовується для збереження зашифрованих даних. Відразу ж записується IV в `memoryStream` за допомогою `memoryStream.Write(aes.IV, 0, aes.IV.Length)`.

Потім створюється `CryptoStream` (`cryptoStream`), який використовує `memoryStream` та `encryptor` для шифрування даних під час записування. Викликаючи `cryptoStream.Write(messageBytes, 0, messageBytes.Length)`, дані `messageBytes` шифруються і записуються у `memoryStream`. Після завершення шифрування дані з `memoryStream` перетворюються в масив байтів за допомогою `ToArray()` і присвоюються змінній `encryptedMessage`.

Отриманий `encryptedMessage` містить зашифроване повідомлення, готове до передачі або подальшого використання.

У завершення зберігається звіт у вигляді .txt файла рисунок 4.10.

```
// Зберігаємо звіт в текстовий файл
string report = "Decryption Key: " + BitConverter.ToString(decryptionKey).Replace("-", "") + Environment.NewLine;
report += "Status: Successful" + Environment.NewLine;
report += "Time: " + DateTime.Now.ToString() + Environment.NewLine;
report += "Recipient Bluetooth Address: " + recipientAddress + Environment.NewLine;
report += "Encrypted Message: " + BitConverter.ToString(encryptedMessage).Replace("-", "") + Environment.NewLine;
report += "Decrypted Message: " + message;

string desktopPath = Environment.GetFolderPath(Environment.SpecialFolder.Desktop);
string reportFilePath = Path.Combine(desktopPath, "decryption_report.txt");
File.WriteAllText(reportFilePath, report);

Console.WriteLine("Report saved to: " + reportFilePath);
```

Рисунок 4.10 –Структура звіту та шлях куди він був збережений

Тепер розглянемо роботу додатків, а потім їх звіти. Додатки не можуть працювати окремо, тому запускається додаток BlueToothSTART. При роботі першого

додатку треба ввести цільову адресу натиснути “enter”, потім запустити другий додаток BlueToothEND яка автоматично приймає та одразу відправляє IoT повідомлення рисунок 4.11.

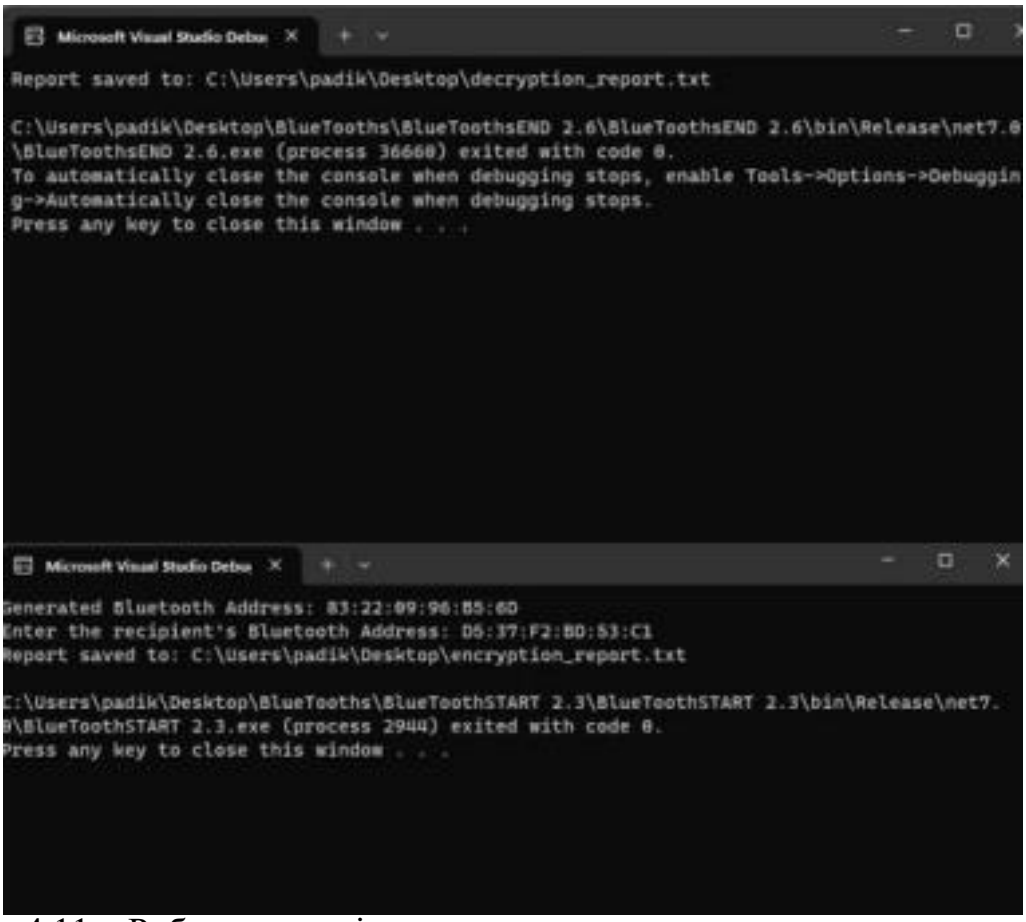

Рисунок 4.11 – Робота додатків

Тепер розглянемо звіти додатків рисунок 4.12-13.

Рисунок 4.12 – На цьому зображенні ми бачимо звіт додатку BlueToothSTART

Звіт дає зрозуміти що з’єднання було встановлене, згенерована та зашифрована адреса була відправлена. Повідомлення від IoT було отримане(на згенеровану адресу)

та розшифроване.

Рисунок 4.13 – На цьому зображенні ми бачимо звіт додатку BlueToothEND

Звіт дає зрозуміти що з’єднання було встановлене, зашифрована адреса була отримана та розшифрована. Потім на отриману адресу було відправлено зашифроване повідомлення [25].

На цьому вважаю мету своєї роботи виконаною, а цілі досягнутими.

4.2 Висновки щодо ефективності застосування механізмів захисту в цілому

У даній дипломній роботі було розроблено додаток, призначений для генерації та зміни BlueTooth-адреси з метою підвищення безпеки системи BlueTooth IoT. Основною метою роботи було дослідити ефективність застосування механізмів захисту у контексті зміни BlueTooth-адреси та оцінити їх вплив на забезпечення конфіденційності та захисту пристроїв IoT від потенційних атак та ідентифікації.

Додаток був розроблений на мові програмування C# це дозволило створити функціональний і надійний інструмент для зміни BlueTooth-адреси пристроїв. У рамках дипломної роботи було проведено широкий огляд наукової літератури та аналіз сучасних підходів до захисту систем IoT, зокрема пристроїв, що використовують технологію BlueTooth. Для досягнення поставленої мети були використані різноманітні механізми захисту, такі як зміна BlueTooth-адреси, шифрування даних.

Ці механізми були інтегровані в розроблений додаток, забезпечуючи його стійкість до атак та підвищуючи рівень безпеки пристроїв IoT. У цьому висновку будуть розглянуті результати проведених досліджень та експериментів, які

підтверджують ефективність механізмів захисту та їх вплив на безпеку системи BlueTooth IoT. Першим важливим механізмом була зміна BlueTooth-адреси. Цей механізм дозволяє пристрою генерувати та змінювати свою унікальну адресу, що ускладнює можливість ідентифікації пристрою та зменшує ризик викриття конфіденційної інформації.

Застосування зміненої BlueTooth-адреси допомагає зберегти конфіденційність користувача та запобігти потенційним атакам, спрямованим на стеження за пристроєм. Крім того, були використані механізми шифрування даних які забезпечують конфіденційність інформації, переданої по BlueTooth, тим самим захищаючи її. У результаті проведених досліджень та експериментів було встановлено, що застосування цих механізмів значно підвищує безпеку системи BlueTooth IoT. Вони сприяють уникненню атак, забезпечують конфіденційність інформації та захищають пристрої IoT від зловмисних дій. Продовження роботи над розробленим додатком та механізмами захисту може включати вдосконалення алгоритмів авторизації, розширення можливостей контролю доступу та підтримку більш широкого спектру BlueTooth-протоколів. Це дозволить ще більш ефективно захищати систему BlueTooth IoT від потенційних загроз та забезпечити високий рівень безпеки для користувачів. Загалом, результати досліджень підтверджують, що застосування розробленого додатку та використання механізмів захисту, зокрема зміни BlueTooth-адреси, шифрування даних, суттєво підвищує безпеку системи BlueTooth IoT та сприяє забезпеченню приватності та захисту пристроїв IoT.

Під час проведення досліджень та експериментів з розробленим додатком для генерації та зміни BlueTooth-адреси, було отримано цінні результати, що свідчать про ефективність використаних механізмів захисту та їх вплив на безпеку системи BlueTooth IoT. По-перше, було виявлено, що зміна BlueTooth-адреси дозволяє значно ускладнити процес ідентифікації пристрою та знизити ймовірність зламу з'єднання чи стеження за користувачем. Генерація нової адреси при підключенні до мережі BlueTooth забезпечує анонімність пристрою та унеможливлює його ідентифікацію на основі постійної адреси. Дослідження також підтвердили ефективність механізмів шифрування даних, використання сильних алгоритмів шифрування дозволяє захистити передані дані від несанкціонованого доступу та забезпечити їх

конфіденційність. Проведені експерименти підтвердили, що застосування цих механізмів значно знижує ризик викриття конфіденційної інформації та забезпечує високий рівень безпеки для пристроїв IoT, що використовують BlueTooth. Виявлено, що пристрої, що використовують розроблений додаток, були надійно захищені від потенційних атак та зловмисних дій. Отже, аналіз результатів досліджень вказує на успішність застосування розробленого додатку та механізмів захисту в контексті зміни BlueTooth-адреси та підвищення безпеки системи BlueTooth IoT. Ці результати підтверджують важливість та доцільність впровадження таких механізмів для забезпечення безпеки та приватності пристроїв IoT.

ВИСНОВКИ

Захист складових системи Інтернету речей є важливим завданням, яке вимагає здійснення комплексних заходів для забезпечення безпеки та захисту від потенційних кібератак. Механізми захисту повинні включати в себе заходи на рівні апаратної та програмної частини системи, захист від шкідливих програм, криптографічні методи захисту даних, аутентифікацію та авторизацію користувачів, мережеву безпеку, захист від DDoS-атак та інші. Для забезпечення високого рівня захисту важливо використовувати сучасні методи та технології, такі як блокчейн, машинне навчання та штучний інтелект. Крім того, важливо бути відкритим до виявлення та усунення потенційних вразливостей в системі, що дозволить зменшити ризик кібератак та підвищити загальний рівень безпеки. Отже, можна зробити висновок, що захист складових системи IoT є складним, але надзвичайно важливим завданням, яке потребує постійного вдосконалення та використання сучасних методів та технологій.

Тільки таким чином можна забезпечити безпеку та захист даних у цій сфері, яка є дедалі більш важливою для суспільства в цілому. Інтернет речей - швидкорозвиваюча галузь, що з'єднує пристрої та датчики в різних сферах. Проте, збільшення пристроїв і згенерованих даних призводить до зростання кібератак і порушень безпеки. Механізми захисту складових системи Інтернету речей повинні бути розглянуті на кожному етапі життєвого циклу системи, починаючи від розробки та виробництва, а закінчуючи експлуатацією та відновленням. Важливо враховувати фактор безпеки на кожному етапі та забезпечувати дотримання стандартів безпеки та конфіденційності даних. Окрім того, механізми захисту повинні бути зорієнтовані на виявлення та реагування на потенційні загрози в режимі реального часу. Це може бути досягнуто за допомогою використання машинного навчання та штучного інтелекту, які можуть аналізувати великі обсяги даних та виявляти аномалії та незвичну поведінку в мережі.

Загалом, забезпечення безпеки та захисту даних у складових системи Інтернету речей є складним завданням, яке потребує постійного вдосконалення та

використання новітніх технологій. Важливо розуміти, що захист від кібератак та порушень безпеки є невід'ємною частиною розвитку Інтернету речей та майбутнього розвитку суспільства в цілому.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Моделювання розумного будинку в середовищі Cisco Packet Tracer – Київ: Хома В.В., Кеньо Г.В., 2018.

2. Що таке розумний будинок і навіщо він потрібен? [Електронний ресурс] // stylus.ua – Режим доступу до ресурсу: https://stylus.ua/uk/articles/528.html. 3. Система автоматизації роботи інженерних систем та окремих приладів будинку [Електронний ресурс] // domos.ua/. – 2019. – Режим доступу до ресурсу: https://domos.ua/.

4. Що таке розумний будинок: функції, види, складові та екосистеми [Електронний ресурс] // ek.ua. – 2018. – Режим доступу до ресурсу: https://ek.ua/ua/post/1990/618-chto-takoe-umnyy-dom-funkcii-vidy-sostavlyayuschie-i ekosistemy/.

5. Єрохін С. Д. Штучний інтелект для інформаційної безпеки. В:2020 Системи генерування та обробки сигналів у сфері бортового зв'язку. IEEE, 2020.с.1-4 6. Системи безпеки розумного будинку [Електронний ресурс] // exposervice. – 2020. – Режим доступу до ресурсу: https://exposervice-p.com.ua/sistemi-bezpeki rozumnogo-budinku/

7. User Experience Design for the Internet of Things – Boston: Claire Rowland, 2017. – 44 с.

8. Analyzing the Impacts of Emerging Technologies on Workforce Skills: A Case Study of Industrial Engineering in the Context of the Industrial Internet of Things [Електронний ресурс] // University of Windsor. – 2020. – Режим доступу до ресурсу: https://scholar.uwindsor.ca/cgi/viewcontent.cgi?article=9472&context=etd.

9. An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things [Електронний ресурс] // IEEE Internet of Things Journal. – 2018. – Режим доступу до ресурсу: https://www.researchgate.net/publication/327855183_An_Ensemble_Intrusion_Detection_

Technique_Based_on_Proposed_Statistical_Flow_Features_for_Protecting_Network_Traf

fic_of_Internet_of_Things.

10. Design, Launch, and Scale IoT Services – California: Apress Berkeley, 2018. – 276 с.

11. Privacy and the Internet of Things [Електронний ресурс] // O'Reilly Media, Inc.. – 2016. – Режим доступу до ресурсу: https://www.oreilly.com/library/view/privacy and-the/9781492042822/.

12. Brian Russell, Drew Van Duren [Електронний ресурс] // Packt. – 2018. – Режим доступу до ресурсу: https://www.researchgate.net/profile/Ali_Al Qurabat2/post/Can_any_one_share_me_a_Book_of_Practical_Internet_of_Things_Securit y/attachment/5a415ea84cde266d587da2a7/AS:575551220207619@1514233512763/down load/Brian+Russell%2C+Drew+Van+Duren-Practical+Internet+of+Things+Security Packt+Publishing+%282016%29.pdf.

13. Demystifying Internet of Things Security [Електронний ресурс] // Apress. – 2019. – Режим доступу до ресурсу: https://library.oapen.org/viewer/web/viewer.html?file=/bitstream/handle/20.500.12657/22 840/1007321.pdf?sequence=1&isAllowed=y.

14. Practical Industrial Internet of Things Security [Електронний ресурс] // Packt. – 2018. – Режим доступу до ресурсу: https://www.packtpub.com/product/practical industrial-internet-of-things-security/9781788832687.

15. IoT Security and Privacy Paradigm [Електронний ресурс] // dokumen. – 2020. – Режим доступу до ресурсу: https://dokumen.pub/iot-security-and-privacy paradigm-internet-of-everything-ioe-1nbsped-0367253844-9780367253844.html.

16. Security and Privacy in Internet of Things (IoTs) [Електронний ресурс] // CRC Press. – 2016. – Режим доступу до ресурсу: https://www.taylorfrancis.com/books/edit/10.1201/b19516/security-privacy-internet things-iots-fei-hu.

17. Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review [Електронний ресурс] // MDPI. – 2023. – Режим доступу до ресурсу: https://www.mdpi.com/2076-3417/13/5/3183

18. П’ять ключових переваг шифрування даних [Електронний ресурс] // Corewin. – 2023. – Режим доступу до ресурсу: https://corewin.ua/blog/five-key-benefits of-data-encryption/.

19. Шифрування: типи і алгоритми. Що це, чим відрізняються і де використовуються? [Електронний ресурс] // Hostpro. – 2020. – Режим доступу до ресурсу: https://hostpro.ua/wiki/ua/security/encryption-types-algorithms.

20. Основні методи шифрування та дешифрування інформації: історичні аспекти [Електронний ресурс] // Полтавська політехніка імені Юрія Кондратюка. – 2016. – Режим доступу до ресурсу: http://reposit.nupp.edu.ua/bitstream/PoltNTU/11072/1/74-%D1%82%D0%B0%20%D0%BA%D0%BE%D0%BD%D1%84%20%D0%A2.1-207- 208.pdf.

21. Технології інтернту речей навчальний посібник [Електронний ресурс] // КПІ ім. Ігоря Сікорського. – 2021. – Режим доступу до ресурсу: https://ela.kpi.ua/bitstream/123456789/42078/1/Zhurakovskyi_B_Zeniv_Tehnologii_internet_rechey.pdf.

22. Вступ до Інтернету речей [Електронний ресурс] // learn.ztu.edu.ua. – 2019. – Режим доступу до ресурсу: https://learn.ztu.edu.ua/pluginfile.php/68838/mod_resource/content/2/%D0%9B-1.pdf.

23. Сучасні методи шифрування інформації [Електронний ресурс] // CORE. – 2014. – Режим доступу до ресурсу: https://core.ac.uk/download/pdf/324233768.pdf.

24. Шифрування даних: все, про що ви повинні знати, щоб захистити дані [Електронний ресурс] // SIM-Network. – 2019. – Режим доступу до ресурсу: https://www.sim-networks.com/ukr/blog/data-encryption-best-practices.

25. Захист інформаційних ресурсів: криптографічні та стеганографічні методи захисту даних [Електронний ресурс] // НПУ імені М.П.Драгоманова Інститут інформатики. – 2012. – Режим доступу до ресурсу: https://vfranchuk.fi.npu.edu.ua/images/files/statty/32_ZIR_cript.pdf.

ДОДАТОК А

Лістинг А.1 Фрагмент програмного коду файлу

BlueToothSTART 2.3.sln

```
using System;
using System.IO;
using System.Text;
using System.Security.Cryptography;

class Program
{
 static string GenerateBluetoothAddress()
 {
 // Генеруємо випадковий Bluetooth адресу
 var random = new Random();
 var addressBytes = new byte[6];
 random.NextBytes(addressBytes);
 addressBytes[0] |= 0x01; // Встановлюємо локальний біт (LSB) 1

 // Перетворюємо адресу на рядок у форматі «XX:XX:XX:XX:XX:XX» var addressBuilder
= new StringBuilder();
 foreach (var b in addressBytes)
 {
 addressBuilder.AppendFormat(«{0:X2}:», b);
 }
 addressBuilder.Length -= 1; // Видаляємо останній двокрапку return
addressBuilder.ToString();
 }

 static byte[] EncryptMessage(string message, byte[] key) {
 using (var aes = Aes.Create())
 {
 aes.Key = key;
 aes.GenerateIV();
 byte[] encryptedData;

 using (var encryptor = aes.CreateEncryptor())
 {
 using (var memoryStream = new MemoryStream()) {
 using (var cryptoStream = new CryptoStream(memoryStream, encryptor, CryptoStreamMode.Write))
 {
 byte[] messageBytes = Encoding.UTF8.GetBytes(message); cryptoStream.Write(messageBytes, 0,
messageBytes.Length);
 }
                                encryptedData = memoryStream.ToArray();

 }
 }
```

```
 byte[] encryptedMessage = new byte[aes.IV.Length + encryptedData.Length];
 Array.Copy(aes.IV, encryptedMessage, aes.IV.Length); Array.Copy(encryptedData, 0,
encryptedMessage, aes.IV.Length, encryptedData.Length);

 return encryptedMessage;
 }
 }

 static void Main()
 {
 // Генеруємо Bluetooth адресу
 string bluetoothAddress = GenerateBluetoothAddress(); Console.WriteLine(«Generated Bluetooth
Address: « + bluetoothAddress);

 // Отримуємо Bluetooth адресу отримувача від користувача Console.Write(«Enter
the recipient’s Bluetooth Address: «); string recipientAddress = Console.ReadLine();

 // Статичний ключ для шифрування (256 біт)
 byte[] encryptionKey = new byte[] {
 0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF,
 0xFE, 0xDC, 0xBA, 0x98, 0x76, 0x54, 0x32, 0x10,
 0x31, 0x2C, 0x5F, 0x7E, 0x12, 0x4A, 0x6D, 0x3F,
 0x8B, 0x9D, 0xC2, 0x3E, 0x5A, 0x17, 0x76, 0xE9
 };

 // Шифруємо Bluetooth адресу
 byte[] encryptedAddress = EncryptMessage(bluetoothAddress, encryptionKey);

 // Передаємо зашифровану адресу до другої програми
 byte[] receivedMessage = encryptedAddress

 // Розшифровуємо повідомлення
 string decryptedMessage = DecryptMessage(receivedMessage, encryptionKey);

 // Зберігаємо звіт в текстовий файл
 string report = «Encryption Key: « +
BitConverter.ToString(encryptionKey).Replace(«-«, «») + Environment.NewLine; report += «Status:
Successful» + Environment.NewLine; report += «Time: « + DateTime.Now.ToString() +
Environment.NewLine; report += «Original Bluetooth Address: « + bluetoothAddress +
Environment.NewLine;
 report += «Encrypted Bluetooth Address: « +
BitConverter.ToString(encryptedAddress).Replace(«-«, «») + Environment.NewLine; report +=
«Recipient Bluetooth Address: « + recipientAddress + Environment.NewLine;
 report += «Received Message: « +
BitConverter.ToString(receivedMessage).Replace(«-«, «») + Environment.NewLine; report +=
«Decrypted Message: « + decryptedMessage;


 string desktopPath =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop); string
reportFilePath = Path.Combine(desktopPath,
«encryption_report.txt»);
 File.WriteAllText(reportFilePath, report);

 Console.WriteLine(«Report saved to: « + reportFilePath); }

 static string DecryptMessage(byte[] encryptedMessage, byte[] key) {
```

```
using (var aes = Aes.Create())
{
aes.Key = key;

byte[] iv = new byte[aes.BlockSize / 8];
byte[] cipherText = new byte[encryptedMessage.Length – iv.Length];

Array.Copy(encryptedMessage, iv, iv.Length);
Array.Copy(encryptedMessage, iv.Length, cipherText, 0, cipherText.Length);

aes.IV = iv;

using (var decryptor = aes.CreateDecryptor())
{
using (var memoryStream = new MemoryStream(cipherText)) {
using (var cryptoStream = new CryptoStream(memoryStream, decryptor, CryptoStreamMode.Read))
{
using (var reader = new StreamReader(cryptoStream)) {
return reader.ReadToEnd();
}
}
}
}
}
}
}
```

Лістинг А.2 Фрагмент програмного коду файлу

BlueToothEND 2.6.sln

```
using System;
using System.IO;
using System.Text;
using System.Security.Cryptography;
class Program
{
 static byte[] DecryptMessage(byte[] encryptedMessage, byte[] key) {
 byte[] iv = new byte[16];
 byte[] encryptedData = new byte[encryptedMessage.Length - iv.Length];

 Array.Copy(encryptedMessage, iv, iv.Length);
 Array.Copy(encryptedMessage, iv.Length, encryptedData, 0, encryptedData.Length);

 using (var aes = Aes.Create())
 {
 aes.Key = key;
 aes.IV = iv;

 using (var descriptor = aes.CreateDecryptor())
 {
 using (var memoryStream = new MemoryStream()) {
 using (var cryptoStream = new CryptoStream(memoryStream, descriptor, CryptoStreamMode.Write))
 {
 cryptoStream.Write(encryptedData, 0, encryptedData.Length);
 }
                                    return memoryStream.ToArray();
 }
 }
 }
 }

 static void Main()
 {
 // Встановлюємо адресу Bluetooth-отримувача
 string recipientAddress = "D5:37:F2:BD:53:C1";

 // Відправка зашифрованого повідомлення (по Bluetooth) byte[] messageBytes =
Encoding.UTF8.GetBytes("the connection is hidden");

 // Статичний ключ для шифрування (256 біт)
 byte[] decryptionKey = new byte[] {
 0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF,
 0xFE, 0xDC, 0xBA, 0x98, 0x76, 0x54, 0x32, 0x10,
 0x31, 0x2C, 0x5F, 0x7E, 0x12, 0x4A, 0x6D, 0x3F,
 0x8B, 0x9D, 0xC2, 0x3E, 0x5A, 0x17, 0x76, 0xE9
 };
```

BlueTooth

```
 // Шифруємо повідомлення
 byte[] encryptedMessage;
 using (var aes = Aes.Create())
```

```
{
aes.Key = decryptionKey;
aes.GenerateIV();

using (var encryptor = aes.CreateEncryptor())
{
using (var memoryStream = new MemoryStream()) {
memoryStream.Write(aes.IV, 0, aes.IV.Length); using (var cryptoStream = new
                                CryptoStream(memoryStream,
encryptor, CryptoStreamMode.Write))
{
cryptoStream.Write(messageBytes, 0, messageBytes.Length);
}
                                encryptedMessage = memoryStream.ToArray();
}
}
}

// Розшифровуємо повідомлення
byte[] decryptedMessage = DecryptMessage(encryptedMessage, decryptionKey);

// Конвертуємо розшифроване повідомлення в рядок
string message = Encoding.UTF8.GetString(decryptedMessage);


// Зберігаємо звіт в текстовий файл
string report = "Decryption Key: " +
BitConverter.ToString(decryptionKey).Replace("-", "") + Environment.NewLine; report += "Status:
Successful" + Environment.NewLine; report += "Time: " + DateTime.Now.ToString() +
Environment.NewLine; report += "Recipient Bluetooth Address: " + recipientAddress +
Environment.NewLine;
report += "Encrypted Message: " +
BitConverter.ToString(encryptedMessage).Replace("-", "") + Environment.NewLine; report +=
"Decrypted Message: " + message;

string desktopPath =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop); string
reportFilePath = Path.Combine(desktopPath,
"decryption_report.txt");
File.WriteAllText(reportFilePath, report);

Console.WriteLine("Report saved to: " + reportFilePath); }
}
```