Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи захисту інформації в IoT

(тема)

Виконав:
студент II курсу, групи КСМзм-19-1
Польська Б.Ю.
(прізвище, ініціали)

Спеціальність
123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Комп'ютерні системи та мережі
(повна назва освітньої програми)

Керівник: доц. Голубничий Д.Ю.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерні системи та мережі
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20___ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Польській Богдані Юріївні
(прізвище, ім'я, по батькові)

1. Тема роботи Методи захисту інформації в IoT

затверджена наказом по університету від " 23 " жовтня 2020 р. № 168 Стз

2. Термін подання студентом роботи до екзаменаційної комісії 14 грудня 2020 р.

3. Вхідні дані до роботи
система управління безпекою
розумний будинок
методи захисту пристроїв
IoT

4. Перелік питань, що потрібно опрацювати в роботі
аналіз предметної області
методи захисту пристроїв в IoT
розробка системи управління безпекою в IoT
модель захисту пристроїв в IoT

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайд презентація – 16 слайдів.

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз завдання	27.10.2020-01.11.2020	
2	Аналіз науково-технічної літератури	02.11.2020-08.11.2020	
3	Пошук моделі загроз системи	09.11.2020-13.11.2020	
4	Пошук аналітичних моделей IoT	14.11.2020-17.11.2020	
5	Вивчення концепції імітаційного моделювання	18.11.2020-25.11.2020	
6	Визначення вторгнень	26.11.2020-29.11.2020	
7	Оформлення пояснювальної записки	30.11.2020-07.12.2020	
8	Оформлення графічної частини	08.12.2020-11.12.2020	
9	Представлення магістерської роботи	12.12.2020-13.12.2020	
	науковому керівнику		

Дата видачі завдання 26 жовтня 2020 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Голубничий Д.Ю.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 91 с., 29 рис., 3 табл., 1 дод., 16 джерел.

МОДЕЛЬ, СИСТЕМА, ІОТ, МЕТОД, МЕХАНІЗМ, КОНФІДЕНЦІЙНІСТЬ.

Метою атестаційної роботи є дослідження існуючих механізмів та методів захисту інформації в IoT.

У ході виконання атестаційної роботи розглянуті базові концепції IoT. Проведено аналіз існуючих еталонних моделей IoT. Розглянуті існуючі механізми та методи захисту пристроїв в IoT. Значну увагу у роботі приділено політиці безпеки та конфіденційності. Розроблена система управління безпекою в IoT. Запропонована та обгрунтована модульна структура управління безпекою підключених пристроїв до мережі.

ABSTRACT

Master’s thesis: 91 pages, 29 figures, 3 tables, 1 appendices, 16 sources.

MODEL, SYSTEM, IOT, METHOD, MECHANISM, CONFIDENTIALITY.

The major goal of this thesis is the advancement of advanced mechanisms and methods for obtaining information in the IoT.

The attestation robots have seen the basic concepts of the IoT. The analysis of current standard IoT models was carried out. Seeing the smart mechanisms and methods for finding attachments in the IoT. I respect the robot because of the security and confidentiality policy. The IoT security management system has been broken up. The modular structure of the management of the bake-free annexes to the framing is proponated and primed.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AMQP – вдосконалений протокол черги повідомлень (англ., Advanced Message Queuing Protocol)

CoAP – протокол обмежених додатків (англ., Constrained Application Protocol)

IoT – інтернет речей (англ., Internet of Things)

IIoT – промисловий інтернет речей (англ., Industrial Internet of Things)

MQTT – протокол телеметрії черги повідомлень (англ., Message Queuing Telemetry Transport)

PKI – інфраструктура відкритих ключів (англ., Public Key Infrastructure)

RFID – радіочастотна ідентифікація (англ., Radio Frequency IDentification)

XMPP – протокол обміну повідомленнями та присутності (англ., eXtensible Messaging and Presence Protocol)

ВСТУП

В епоху підключених пристроїв, відомих як Інтернет речей, люди користуються послугами, що надаються предметами IoT, речами чи пристроями у всіх аспектах свого життя, включаючи будинок, офіси, машини, лікарні, навіть людські тіла та інші. З появою хмар і широким розгортанням мереж Wi-Fi та IPv6 IoT зростає надзвичайно швидкими темпами. Відповідно до звіту про мобільність, опублікованого компанією Ericsson, кількість пристроїв IoT до 2022 року буде вражаюче збільшена до приблизно 18 млрд. Підключені до IoT пристрої, як правило, працюють з низьким енергоспоживанням і сильно обмежені, коли йдеться про пам’ять, обробну потужність, заряд акумулятора, простір тощо. Ці обмежені пристрої, як правило, базуються на мікроконтролерах, що мають обмежений набір функціональних можливостей.

IoT – це нерегульований домен, і досі не прийнято стандартів. Фрагментація стає все більш помітною у процесі розробки IoT. Для того, щоб спеціалісти з програмного забезпечення могли зручно створювати додатки для пристроїв IoT, існує потреба у дефрагментації та сумісній загальній платформі загального програмного забезпечення з відкритим кодом. Відкритість у IoT - це відкриті стандарти та сумісність, спільноти з відкритим кодом, відкритий доступ до даних досліджень, відкритий доступ до спільних даних користувачів, відкритий API тощо. Відкриті дані та Open API потрібні для того, щоб передавати дані, зібрані датчиками, спільноті розробників додатків. Незважаючи на те, що відкриті системи пропонують безліч переваг для розробки програмного забезпечення, постачальник несе більшу відповідальність за використання рішення з відкритим кодом для забезпечення безпеки та конфіденційності пристрою IoT.

Хоча використання відкритих систем IoT має багато переваг, проте безпека не гарантується. Вразливі місця безпеки можуть призвести до

порушень конфіденційності, грошових втрат, шкідливого програмного забезпечення та програм-вимог, інших створених атак безпеки, навіть втрати життя, наприклад шляхом контролю зламаних автономних транспортних засобів. Сума точок атаки, де зловмисний користувач може спробувати атакувати систему IoT, тобто поверхня атаки або вектор атаки зростає із збільшенням кількості підключених пристроїв. У системах IoT відбувається величезна кількість передачі даних. Деякі з цих даних є особистими, наприклад інформація про здоров'я, контактна інформація, IP-адреса, місцезнаходження GPS тощо, а деякі мають таємний характер, наприклад реквізити банківських рахунків, облікові дані тощо. Це передбачає гостру потребу у безпеці даних та механізмах конфіденційності даних. Існуючі методи управління безпекою та конфіденційністю можуть бути не завжди застосовними через різні відмінності між системами IoT та застарілими системами. Таким чином, вирішення проблем безпеки та конфіденційності в Інтернеті речей є надзвичайно важливим завданням, хоча цього може бути важко досягти. Працівники IoT зазвичай ігнорують безпеку з різних причин, наприклад, через відсутність досвіду в галузі безпеки, економію витрат та обмеження часу тощо. Практикуючий IoT застосовує реактивний підхід до пом'якшення проблем безпеки, замість того, щоб активно вбудовувати безпеку в продукт із самого початку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 IoT – internet of things. Інтернет речей

Концепція обчислювальної мережі фізичних предметів («речей»), оснащених вбудованими технологіями для взаємодії один з одним або з зовнішнім середовищем, яка розглядає організацію таких мереж як явище, здатне перебудувати економічні та суспільні процеси, що виключає з частини дій і операцій необхідність участі людини.

Поєднуючи підключені пристрої з автоматизованими системами, можна "зібрати інформацію, проаналізувати її та створити дію", щоб допомогти комусь із певним завданням або навчитися з процесу. Насправді це варіюється від розумних дзеркал до маяків у магазинах та поза ними.

По одному з визначень, з точки зору IoT, «річ» - будь-який реальний чи віртуальний об'єкт, який існує і переміщується в просторі і часі і може бути однозначно визначений. Тобто інтернет речей [1] – це не просто безліч різних приладів і датчиків, об'єднаних між собою дротяними і бездротовими каналами зв'язку і підключених до мережі Інтернет, а це більш тісна інтеграція реального та віртуального світів, в якому спілкування виробляється між людьми і пристроями.

З точки зору технологій, IoT є чотириланковою системою: спільні пристрої (сенсори, датчики, термінали), мережі, за якими вони взаємодіють, IoT-платформи і додатки для кінцевих користувачів. При цьому перші два рівня виключити зі структури не можна, платформи в рішенні присутні варіативно, клієнтський інтерфейс поки наявний всюди, але в майбутньому, можливо, рівень програми та інші додаткові елементи в управлінні відпадуть. Взаємодія буде зводитися до backend-додатку, який буде аналізувати дії людини – і на основі цього аналізу формувати взаємодію з кінцевими пристроями без додаткового натискання кнопок. Наприклад, система

«розумного будинку» буде знати, о котрій годині (і з якою яскравістю) хазяїн зазвичай включає світло, на яку годину розігріти вечерю і о котрій потрібно включити телевізор, тому що починається улюблений серіал домовласника. Все це буде працювати на основі шаблонів, що зберігаються в базі системи, і формувати ці дії вона буде без участі людини. Діяльність же розробників і користувачів буде зводитися до контролю – або через додаток, або через механічне виконання дій.

Лампочки разом з холодильниками, кавоварками, мікрохвильовими пічками, дитячими моніторами, охоронними камерами, динаміками, телевізорами та термостатами за останні кілька десятиліть перетворилися із звичайних предметів у канали для майбутнього. Вбудовані сенсорами, які бачать, чують та торкаються навколишнього світу, вони можуть перетворити фізичну інформацію в цифрові дані. У сукупності ці пристрої - а їх мільярди у всьому світі – складають "Інтернет речей".

Практично все з мережевим підключенням належить до Інтернету речей, від камер безпеки та динаміків до смарт-годинників та джинсових куртки. У "розумному будинку" ці гаджети, що підтримують Інтернет, позбавляють нас від наших завдань, повертають нам частину нашого часу та додають нотку новизни до звичайних вражень. Але Інтернет речей - це не просто використання голосу для попереднього розігрівання духовки або використання телефону для вимкнення вогнів. Інтернет речей робить наше фізичне оточення доступним для наших цифрових комп'ютерів, розміщуючи датчики на всьому світі і переводячи його в цифровий формат. Об'єкти, пов’язані з Інтернетом, можуть стати ключем до розблокування прогнозів щодо всього, від поведінки споживачів до подій клімату, але ті самі об’єкти можуть запросити хакерів у особисті простори та витікати інтимні дані.

Більшість людей не почали будувати екосистему «розумних» пристроїв у своїх будинках до масового прийняття голосового контролю. У 2014 році Amazon представила Echo, динамік з вбудованою голосовою помічницею на ім’я Alexa. Apple представила Сірі, свого власного голосового помічника, за

чотири роки до цього – але Сірі жила на вашому телефоні, а Alexa жила всередині диктора і могла керувати всі «розумні» пристрої у вашому домі. Позиція голосового помічника як центральної частини розумного будинку мала кілька наслідків: це демістифікувало Інтернет речей для споживачів, спонукало їх купувати більше гаджетів з підтримкою Інтернету та заохочувало розробників створювати більше «навичок» або команд.

1.2 Історія IoT

Першою «річчю», пов’язаною з Інтернетом, яка скористалася цим новим протоколом, був тостер. Джон Ромкін створив його для виставки 1990 року Interop, виставки для комп'ютерів. Ромкін кинув кілька тортів хліба в тостер і, використовуючи незграбний комп’ютер, увімкнув тостер. Минуло б десятиліття, перш ніж хтось скористався фразою "Інтернет речей", але чарівний тостер "Рокі" показав, яким може бути світ речей, пов'язаних з Інтернетом. (Зрозуміло, це було не повністю автоматизовано; людині все-таки довелося вводити хліб.) Це була частина трюку, частина доказу концепції - і повний перегляд того, що має бути.

Сам термін "Інтернет речей" був введений в 1999 році, коли Кевін Ештон виклав його в презентації PowerPoint для Procter & Gamble. Ештон, яка тоді працювала в оптимізації ланцюгів поставок, описала систему, коли датчики діяли як очі та вуха комп'ютера – абсолютно новий спосіб комп'ютери бачити, чути, торкатися та інтерпретувати своє оточення.

Коли домашній Інтернет став повсюдним, а Wi-Fi збільшився, мрія про розумний дім стала більше схожа на реальність. Компанії почали впроваджувати все більше і більше таких винаходів: «розумні» кавоварки для заварювання ідеальної чашки, духовки, які випікають печиво з точністю часу, і холодильники, які автоматично поповнювали молоко з простроченим терміном. Перший із них, підключений до Інтернету, холодильник, вийшов на ринок у 2000 році. Він міг зібрати вміст полиць, терміни придатності, та

мав MP3-плеєр. Він коштував 20 000 доларів. Оскільки датчики стали дешевшими, ці пристрої, підключені до Інтернету, стали більш доступними для більшої кількості споживачів. А винахід розумних штепсельних вилок, означав, що навіть звичайні предмети можуть стати «розумними» - або, принаймні, ви можете їх включати та вимикати за допомогою телефону.

Будь-яка система IoT сьогодні містить кілька основних компонентів. По-перше, є річ, оснащена датчиками. Цими датчиками може бути все, що збирає дані, як камера всередині розумного холодильника або акселерометр, який відстежує швидкість в розумному біговому взутті. У деяких випадках датчики поєднуються разом для збору декількох точок даних: термостат Nest містить термометр, але також датчик руху; він може регулювати температуру кімнати, коли відчує, що нікого в ній немає. Щоб зрозуміти ці дані, пристрій має певне підключення до мережі (Wi-Fi, Bluetooth, стільниковий або супутниковий) та процесор, де їх можна зберігати та аналізувати.

Все починалося з локальних рішень, які дозволяли автоматизувати щось шляхом установки зв'язку (провідний або з використанням каналів малої дальності дії) між агрегатами: системи автоматичного оповіщення, автоматичного відкриття і закриття воріт, моніторингу і т.д. Технічно структура була така ж: залізо, що збирає інформацію, яке передає її ззовні і виконує команди, мережі, а також системи, які у себе це з’єднували в єдине ціле, містили бізнес-логіку і шар додатку. Раніше автоматизація була доступна тільки в обмеженій кількості областей і тільки для підприємств з великим бюджетом.

У той же час, в інженерії накопичуються кількісно-якісні зміни, які стають причиною якісних змін в бізнесі. У кожній зі складових майбутньої IoT-архітектури (залізо, мережі, сервера) протягом декількох десятиліть відбувалися значні зміни. Девайси стали менше за габаритами, їх вартість істотно знизилася, у багатьох моделей помітно збільшилися терміни автономної роботи – потужні вбудовані акумулятори тепер дозволяють пристроям працювати до 10 років без підключення до харчування. Ринок

тепер може споживати датчики мільярдами – і застосовувати їх в областях, в яких раніше це було неможливо.

Також з'явилися нові мережі (5G, Lo-Ra, NB-IoT), які можуть забезпечити високу пропускну здатність і енергоефективність обміну даними. Сервери тепер постійно розвиваються, їх потужності (по «закону Мура») раз на два роки кратно збільшують свої потужності, до цього додалися софтверні перетворення, з'явилися IoT-платформи, розвивається Big Data. Все це дає можливість вирішувати нові бізнес-завдання, створювати нові види бізнесу (такі як каршерінг) і, головне, застосовувати все це в компаніях меншого масштабу і з меншими витратами.

Телематика (область інформатики, що охоплює сферу телекомунікацій) до вищезазначених змін представляла собою просте отримання інформації і передачу її на конкретний комп'ютер, тому в свідомості потенційних клієнтів вона асоціюється з чимось архаїчним, дорогим і сильно обмеженим по функціоналу. IoT ж є наступним кроком в ланцюжку після телематики, може виконувати її завдання, – але також включає в себе куди більш широке коло можливостей. У свою чергу, все прогресивне повинно позначатися новим терміном. Згадаймо виробників телевізорів - всі ці Full HD, 4K, Ultra HD і так далі. По суті це просто стадії розвитку технологій якості передачі зображення - але, щоб підкреслити різницю між картинкою в старих і нових моделях, технологіям даються нові назви. Таким чином, термін IoT був створений і популяризував, в першу чергу, щоб захистити новий виток розвитку технологій в очах клієнтів та інвесторів. При цьому концепція IoT не має якихось жорстких рамок і обмежень - при бажанні в Інтернет Речей можна включити навіть роботів на Марсі, керованих космічними центрами НАСА.

Поки що Телематика і IoT ділять один ринок, але поступово процентна частка телематики в ньому буде зменшуватися. При цьому Телематика як така не зникне, оскільки залишаться специфічні клієнти - такі як ізольовані від інтернету військові організації.

На рівні масової свідомості IoT також часто сприймається тільки як

джерело даних для Big Data – тобто як сукупність датчиків і сенсорів, які збирають інформацію звідти, звідки її раніше не могли витягати. При цьому вважається, що кінцева цінність витягується на рівні алгоритмів мереж, а Інтернет Речей помилково розглядають тільки як додаток зі збору та агрегації даних. Однак IoT відрізняється від телематики і Big Data можливістю не тільки збирати інформацію про об'єкт, але і управляти цим самим об'єктом. Основна цінність Internet Of Things - в автоматизації процесів і в можливості управляти чимось на рівні об'єктів реального світу, викресливши з ланцюжка людей-посередників. Хоча при цьому IoT все ж знаходиться в тісному зв'язку з нейронними мережами і Big Data, оскільки останні працюють з даними, агреговані IoT і телематикою.

1.3 Як керувати пристроями Internet of Things

Щоб функціонувати за призначенням, пристроями IoT потрібно керувати як внутрішньо (наприклад, технічне обслуговування програмного забезпечення), так і зовні (тобто їх зв'язок з іншими пристроями).

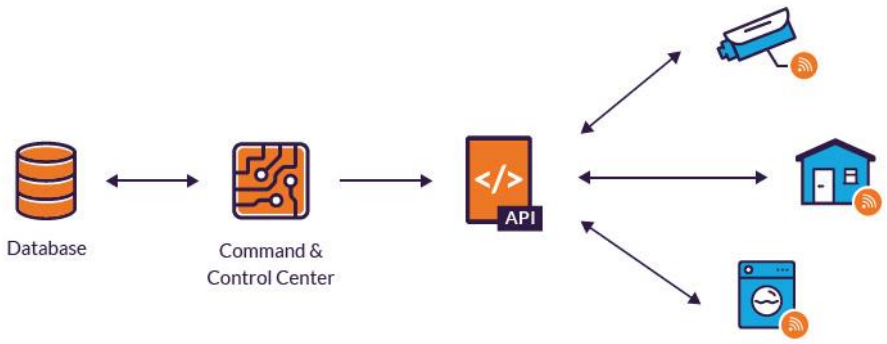

Рисунок 1.1 – Зв’язок пристроїв

Це досягається підключенням кожного пристрою IoT до блоку управління, відомого як центр управління та управління (C&C). Центри відповідають за технічне обслуговування програмного забезпечення,

конфігурації, оновлення мікропрограмного забезпечення для помилок виправлень та вразливості, а також за надання та аутентифікацію завдань, таких як реєстрація пристроїв.

Зв'язок між пристроями [2] вмикається через інтерфейс прикладної програми (API), як зображено на рисунку 1.1. Після того, як виробник пристрою відкриє свій API, інші пристрої або програми можуть використовувати його для збору даних і спілкування. Деякі API навіть дозволяють контролювати пристрої. Наприклад, керівник будівлі може використовувати API для віддаленого блокування дверей всередині певного офісу.

1.4 Industrial Internet of Things, промисловий Інтернет Речей

IIoT означає Індустріальний Інтернет речей або Industrial IoT, який спочатку в основному посилався на промисловий каркас, завдяки якому велика кількість пристроїв або машин підключаються та синхронізуються за допомогою використання програмних засобів та технологій третьої платформи в машині до машини та Інтернеті Контекст речей, пізніше Industrial 4.0 або Індустріальний Інтернет.

Сьогодні IIoT в основному використовується в області застосувань Internet of Things поза споживчим простором та корпоративним ринком IoT, як термін для застосувань та випадків використання у кількох галузях промисловості.

Промисловий Інтернет речей або IIoT визначають як "машини, комп’ютери та люди, які дозволяють інтелектуальні промислові операції, використовуючи розширену аналітику даних для трансформаційних результатів бізнесу".

1.5 Зв'язок машина-машина та IIoT

У чистому контексті "машина до машини" (рисунок 1.2) та "Industrial 4.0" перевага систем полягає в тому, що вони можуть працювати напів незалежно або з дуже мінімальним втручанням людини.

Такі системи дедалі більше зможуть інтелектуально реагувати та навіть змінювати свій хід дій на основі інформації, отриманої за допомогою циклів зворотного зв'язку, встановлених у рамках.

Як вже згадувалося, ключовим словом тут є комунікація між машиною (M2M), яка є елементом Інтернету речей, але також відноситься до конкретної діяльності та початкових етапів Індустріального Інтернету речей.

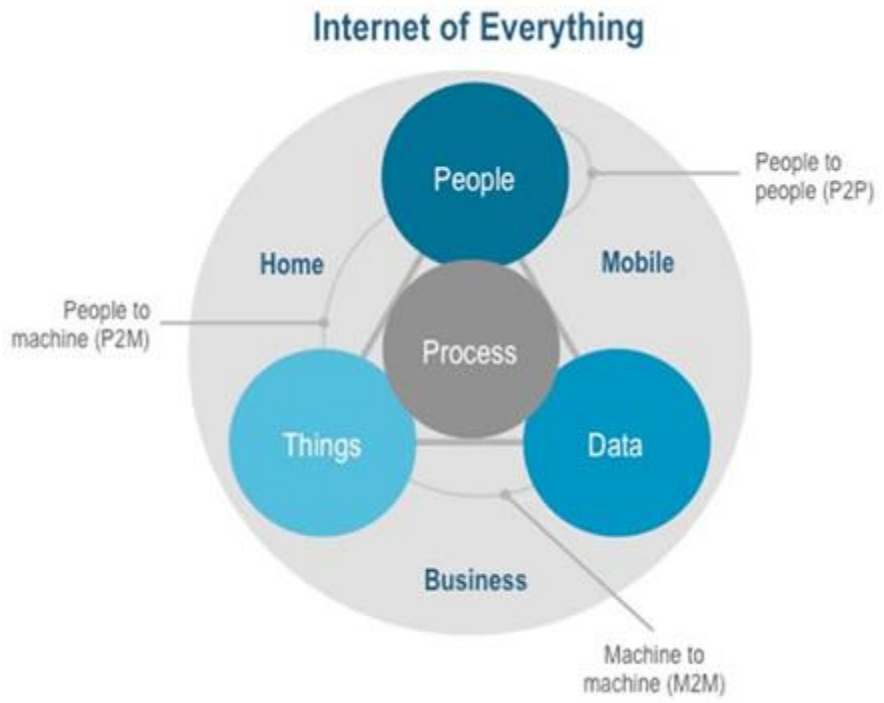

Рисунок 1.2 – Місце від машини до машини або M2M в Інтернеті

IIoT у цьому сенсі можна вважати рухом до «розумних машин», завдяки якому рівні точності операцій, що беруть участь у відповідних системах, підвищуються до рівня, якого неможливо досягти втручанням людини.

1.6 Переваги Industrial IoT у виробництві та за його межами

Однією з найбільших переваг Індустріального Інтернету речей слід вважати зменшення помилок людини та ручної праці, підвищення загальної ефективності та зменшення витрат як у часі, так і в грошах. Ми також не можемо забути можливі основи IIoT у контролі та обслуговуванні якості.

Індустріальний Інтернет речей є частиною Інтернету речей. Інтернет речей або IoT багатий на дані: велика кількість даних збирається, агрегується та ділиться змістовно. Тут знову ж таки метою є підвищення рівня автоматизації на внутрішньому та комерційному рівнях. У промисловому Інтернеті речей дані також мають вирішальне значення, і це спричиняє зміну людських завдань у контексті Industry 4.0, завдяки чому автоматизація призводить до зменшення конкретних видів роботи, але в той же час вимагає нових наборів навичок. Мета Індустріального Інтернету речей – також не повністю замінити людську роботу, її мета – посилити та оптимізувати її, наприклад, створивши нові потоки доходів та бізнес-моделі, що мають велику роль для даних (аналізу) [4].

Інтелектуальна установка циклу зв'язку між машинами дозволяє своєчасно звернути увагу на проблеми обслуговування. Рівень безпеки операцій також підвищується за рахунок зменшення факторів ризику.

Промисловий Інтернет речей піднімає переваги Інтернету речей загалом на більш високий рівень, а також для галузей з високими ставками, де людські помилки можуть спричинити великі ризики. Рівень точності, який може бути досягнутий за допомогою IIoT, є однією з найбільших переваг, що робить цю дисципліну одним з найпривітніших подарунків IoT.

Недалекі часи, коли цілі операції та процеси виробництва можуть змусити себе працювати майже незалежно. Більше того, Індустріальний Інтернет речей використовується для багатьох випадків використання, які допомагають нам зменшити вплив робочої сили людини, що завжди матиме значення, до сценаріїв з високою промисловою небезпекою.

У найближчі роки Industrial IoT, ймовірно, змусить використовувати більш уніфіковані протоколи та архітектури пристроїв, що дозволить машинам безперебійно спілкуватися і тим самим підвищити сумісність.

Підсумовуючи, ось деякі ключові переваги IIoT в галузевому контексті:

- поліпшений та розумний зв’язок між пристроями чи машинами;
- підвищення ефективності;
- економія витрат;
- економія часу;
- підвищена промислова безпека.

1.7 IoT-шарова архітектура

Еталонна модель IoT була розділена на чотири основні шари. Шари знизу вгору – це елементний рівень, мережевий рівень, рівень обслуговування та рівень додатків, як показано на рисунку 1.3.

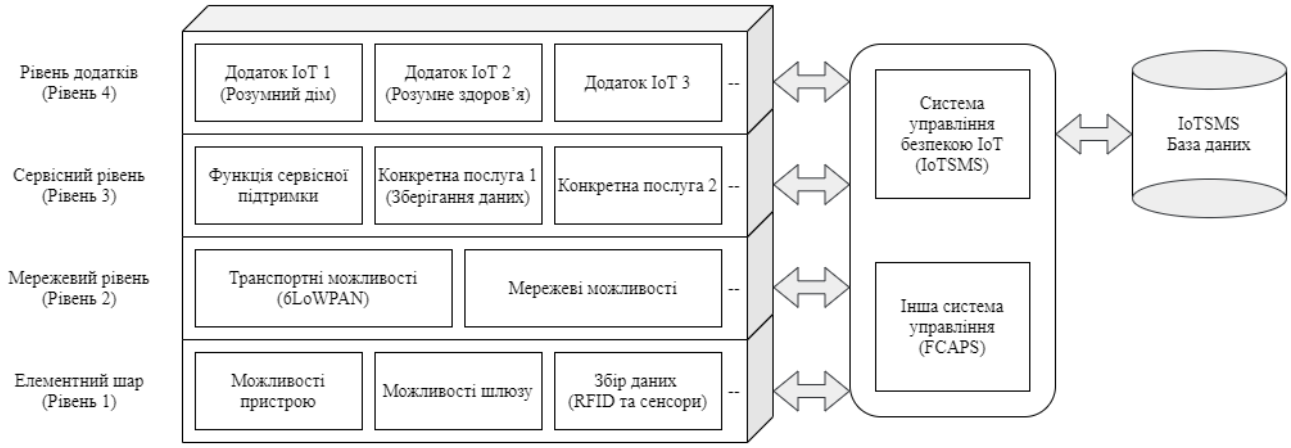

Рисунок 1.3 – Базова еталонна модель IoT

Кожен рівень має свої компоненти [5], стандарти зв'язку та протоколи. Перевагами багатошарової архітектури є:

- забезпечення модульного управління IoT. Ми можемо впровадити

різні протоколи безпеки, служби безпеки та механізми безпеки на кожному рівні, щоб покращити загальний захист мережевої системи IoT;

- шарувата структура легко розширюється, а нижні шари забезпечують обслуговування верхніх шарів;
- дозвіл на те, щоб нові технології як апаратного, так і програмного забезпечення були включені в існуючу мережеву систему IoT, а багаторівневою структурою легко керувати, а також конфігурувати в практичній реалізації.

1.7.1 Елементний шар

Елементний шар – це найнижчий шар із чотирьох шарів IoT. Фактично це рівень пристрою і складається з різних типів вузлів і датчиків, таких як RFID, етикетки зі штрих-кодами, виконавчі механізми та інтелектуальні пристрої виявлення. Ці датчики використовуються для ідентифікації об'єктів, а також для перенесення отриманих даних на наступний шар. Пристрої збирають і завантажують дані на мережевий рівень прямо чи опосередковано. Очікується, що в майбутньому всі пристрої будуть підтримувати IPv6 [6].

Спочатку пристрої, обладнані RFID, можуть використовуватися для моніторингу та контролю стану та контролю доступу в IoT. Радіочастотна ідентифікація (RFID) – це бездротовий пристрій, що використовує електромагнітні поля для передачі даних. Призначенням RFID є ідентифікація та відстеження тегів, прикріплених до об'єктів. Тег зберігає електронну інформацію. Деякі мітки живляться від батареї, інші – від електромагнітної індукції від магнітного поля від зчитувача міток. Тег RFID можна прикріпити до об'єкта та використовувати для управління та відстеження. RFID широко використовується в багатьох додатках. Він пропонує багато переваг перед штрих-кодом, тег можна читати навіть покриваючи іншими об'єктами, він може читати сотні одночасно, а вартість

пасивного тегу починається з 0,09 доларів США кожен. RFID – це один із методів автоматичної ідентифікації та збору даних.

Стандартний протокол IEEE 802.15.4. IEEE 802.15.4 – це стандарт, який визначає вимоги до фізичного рівня та управління доступом до мультимедіа (MAC) для малопотужних бездротових персональних мереж, який був визначений у 2003 році. між пристроями [7].

IEEE 802.15.4 можна використовувати виключно з бездротовою бездротовою персональною мережею на базі IPv6 (6LoWPAN) для побудови бездротової вбудованої мережі для IoT. Основний діапазон зв'язку становить 10 метрів зі швидкістю передачі 250 Кбіт / с. Вищі шари не визначені стандартом. Архітектура базового IEEE 802.15.4 показана на рисунку 1.4.

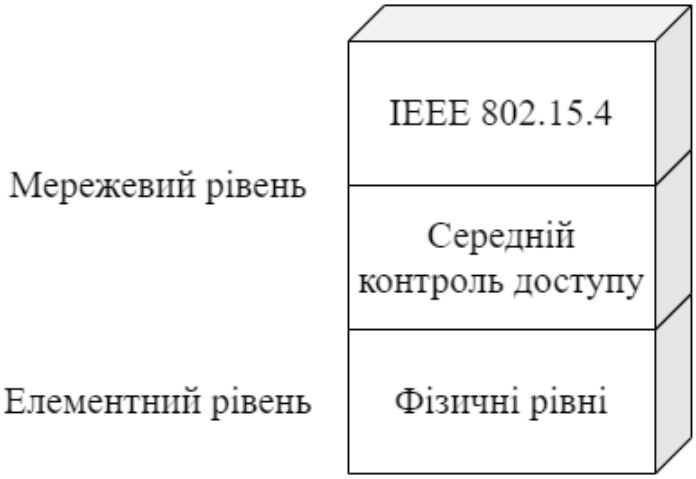

Рисунок 1.4 – Архітектура IEEE 802.15.4.

1.7.2 Мережевий рівень

Мережевий рівень передає дані, отримані від рівня елементів, у верхній рівень. Мережевий рівень передає інформацію за допомогою існуючих методів зв'язку або дротової, або бездротової мережі, Інтернету, хмари, мобільної мережі, супутникової мережі або військової мережі. IoT вимагає масштабованості в мережі великої кількості пристроїв. Щорічно до системи

буде додаватися більше мільярда пристроїв. З цієї причини IPv6 відіграватиме важливу роль в обробці масштабованості мережевого рівня.

Протокол 6LoWPANX. Бездротова персональна мережа IPv6 з низьким енергоспоживанням (6LoWPAN), названа робочою групою Інженерних технологій (IETF), відповідає вимогам пристроїв з низьким енергоспоживанням, а також слабким обчислювальним можливостям вузлів та датчиків, які є основними елементами складають IoT. 6LoWPAN дозволяє отримувати пакети, а також надсилати їх через мережі на базі IEEE 802.15.4.

1.7.3 Сервісний рівень

Сервісний рівень складається з функціональних можливостей, які обробляють зібрані дані та забезпечують посилання на сховище отриманих даних з рівня елементів. Цей рівень IoT служить інтерфейсом між різними пристроями IoT і забезпечує способи зв'язку між елементами. Сервісний рівень поверх мережевого рівня забезпечує зв'язок між датчиками та прикладним рівнем. Він також надає послуги для забезпечення ефективного функціонального зв'язку між додатком та пристроями [8]. На рисунку 1.5 показано, як дані проходять через сервісний рівень як рівень інтеграції.

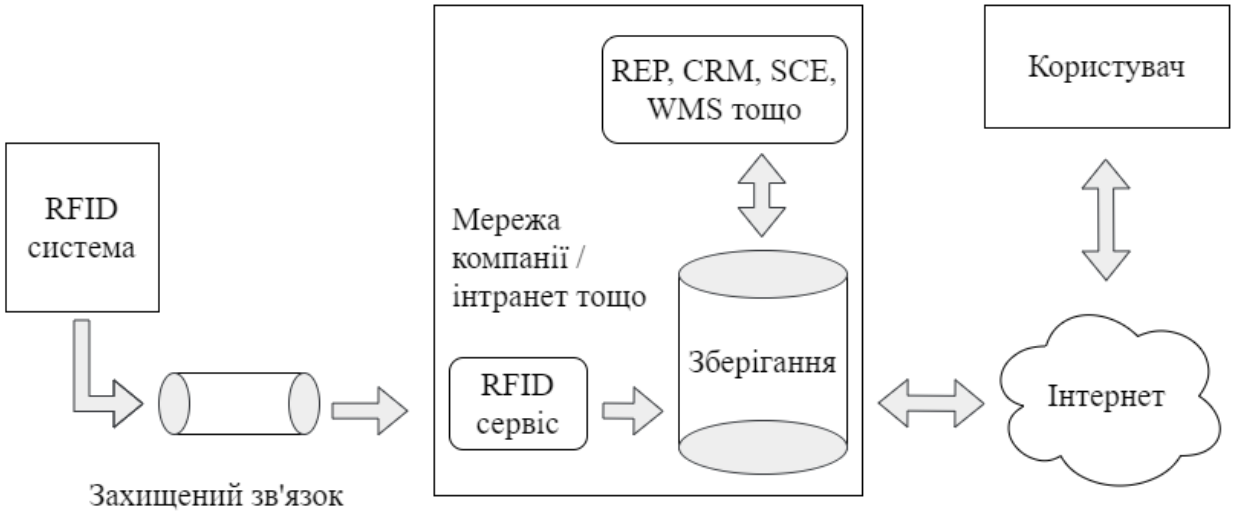

Рисунок 1.5 – Дані проходять через сервісний рівень

Для системи RFID, як приклад реалізації сервісного рівня, ми можемо згадати “Open Remote”, таке як проміжне рішення для житлових та комерційних будівель та автоматизації.

1.7.4 Рівень програми

Прикладний рівень складається з різноманітних практичних застосувань IoT на основі вимог користувачів. Прикладний рівень використовує різну кількість різних протоколів, таких як протокол обмежених додатків (CoAP), протокол телеметрії черги повідомлень (MQTT), вдосконалений протокол черги повідомлень (AMQP) та розширюваний протокол обміну повідомленнями та присутності (XMPP):

- протокол обмежених додатків. Протокол обмежених програм (CoAP) - це синхронний протокол запиту/відповіді, розроблений Інженерною робочою групою Інтернет (IETF). Він був розроблений з використанням підмножини схем HTTP, що робить його сумісним з HTTP. CoAP працює над UDP. Протоколи прикладного рівня на основі UDP зменшують вимоги до пропускної здатності та підтримують багатоадресну та одноадресну передачу, не так, як TCP, який не підтримує багатоадресну передачу. Цільовою метою CoAP є пристрої, обмежені ресурсами, такі як мобільний телефон, планшет, ноутбук та пристрої з низьким енергоспоживання;

- транспортний протокол телеметрії черги повідомлень. Транспорт телеметричної черги повідомлень (MQTT) є протоколом прикладного рівня, MQTT працював від IBM як легкий протокол зв'язку машина-машина (M2M). MQTT працює поверх TCP. Це асинхронний протокол публікації/підписки, який зменшує пропускну здатність мережі та знижує вимоги до обчислень. MQTT розроблений для задоволення вимог щодо низької пропускної здатності та використання батареї. Месенджер Facebook використовує протокол MQTT. MQTT може мати менші затримки, але CoAP отримує низькі втрати пакетів і більшу надійність, надаючи можливість використання

якості обслуговування (QoS). Протокол MQTT використовує функції захисту транспортного рівня (TLS) / рівень захищених сокетів (SSL) так само, як транзакції HTTP через Інтернет;

- розширений протокол черги повідомлень. Розширений протокол черги повідомлень (AMQP) в основному використовується у фінансовій галузі. JPMorgan AMQP використовує 1 мільярд повідомлень на день. AMQP має базову надійність, коли працює через протокол TCP. Він забезпечує асинхронну систему обміну повідомленнями публікації / передплати. Дослідження показують, що рівень успіху зростає із збільшенням пропускної здатності, і в порівнянні з іншими конкурентами AMQP може надсилати більшу кількість повідомлень в секунду. Це забезпечує надійність завдяки гарантіям доставки повідомлень. Безпека в AMQP забезпечується за допомогою протоколу TLS / SSL. У мережевому середовищі IoT все, що розроблено різними постачальниками мереж і виробниками, повинно бути підключене до Інтернету; різні форуми та організації беруть участь у мережевому середовищі IoT [9];

- розширюваний протокол обміну повідомленнями та присутності. Розширюваний протокол обміну повідомленнями та присутності (XMPP) був стандартизований IETF, і він розроблений для спілкування майже в реальному часі та працює через TCP. Протокол XMPP має основну специфікацію захисту TLS / SSL. Однак він не підтримує опції QoS, що робить його недоцільним для зв'язку M2M. Хоча XMPP має свої недоліки, але останнім часом відновив значення як відповідний протокол для IoT.

1.8 Потік даних між шарами

Потік даних IoT можна розділити на чотири фази – збір даних, передача даних, зберігання даних та аналіз даних. На рисунку 1.6 показано потік даних IoT.

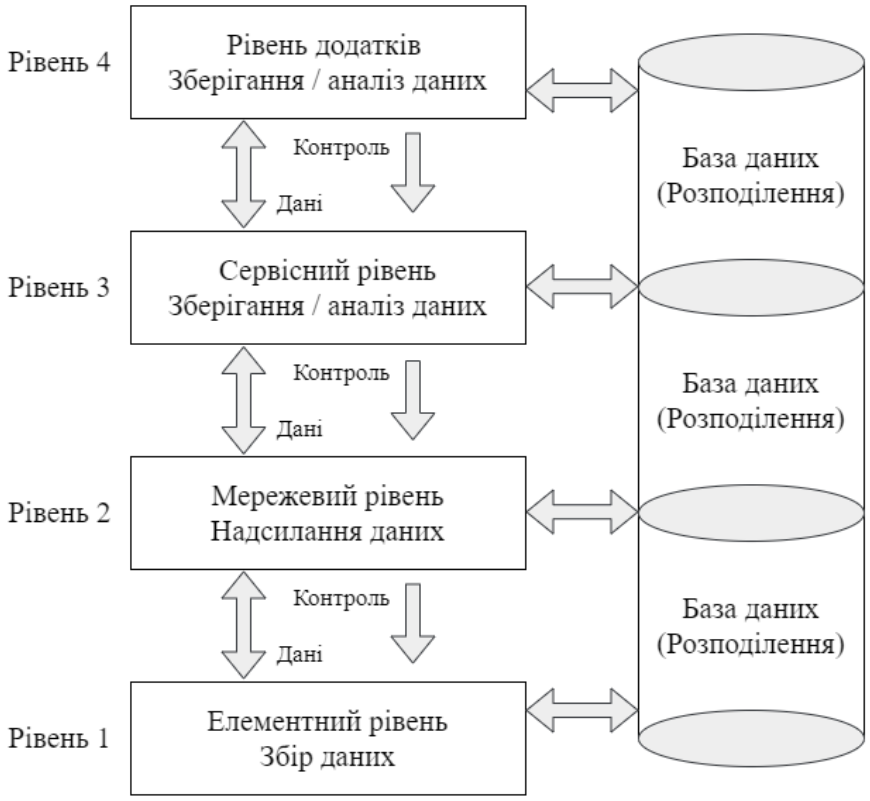

Рисунок 1.6 – Потік даних між шарами

Збір та зберігання даних необхідні для виконання так званих п'яти відомих функцій управління мережею (FCAPS): несправності, конфігурації, обліку, управління продуктивністю та безпекою:

- збір даних. Пристрої елементного шару, що збирають дані з навколишнього середовища; дані надходять з різних інтелектуальних пристроїв, таких як RFID, етикетки зі штрих-кодами, виконавчі механізми та інтелектуальні пристрої виявлення. Ці дані будуть надіслані на верхній рівень безпечними методами. Тип вмісту даних може бути XML на JSON (позначення об’єкта сценарію Java) і залежить від служби HTTP та її конфігурації;

- передача даних. Дані, зібрані розумними пристроями з елементарного рівня, надсилатимуться безпечним способом, щоб гарантувати автентифікацію, цілісність та доступність через мережевий

рівень. Для забезпечення безпеки даних слід впровадити деякі стандарти, такі як стандарт IEEE 802.15.4 та протокол 6LoWPAN [10];

- зберігання даних. Щоб гарантувати надійність та доступність даних, сховище даних потребує резервного копіювання та резервування. Це забезпечує необхідне дублювання даних у випадку відмови системи та у випадку критичних умов;
- аналіз даних. Сервісний рівень забезпечує зв'язок між елементарним рівнем та рівнем програми; він також забезпечує необхідне програмне забезпечення для верхніх шарів. Дані, що передаються через мережевий рівень, повинні бути проаналізовані та реалізовані програмним забезпеченням прикладного рівня.

IoT визначається як мережа інтелектуальних систем для більш пов'язаного світу. Насправді це мережа фізичних об’єктів, які можуть взаємодіяти між собою, обмінюватися інформацією та здійснювати дії. IoT використовує технології RFID та датчиків, дротовий та бездротовий зв’язок, технології низького енергоспоживання, хмарні обчислення та передові Інтернет-протоколи. Довідкова модель IoT влаштована як чотирирівнева функціональна архітектура:

- рівень елементів;
- мережевий рівень;
- рівень обслуговування;
- рівень додатків.

Елементний шар – це найнижчий шар, який включає датчики, виконавчі механізми, пристрої тощо та збирає дані в режимі реального часу. Мережевий рівень – це мережева інфраструктура зв'язку та підтримує вимоги до пропускної здатності та безпеки. Це дозволяє декільком організаціям спільно використовувати та використовувати одне і те ж мережеве середовище самостійно. Сервісний рівень збирає та аналізує періодичні дані безпеки, забезпечує безпеку та витягує відповідну інформацію з величезної кількості необроблених даних. Прикладний рівень забезпечує

користувальницький інтерфейс для використання IoT. Він підтримує різні додатки, такі як охорона здоров'я, транспорт, ланцюги поставок, розумні міста тощо. Основною проблемою для IoT є величезна кількість даних, велика кількість пристроїв та фізичних компонентів, енергоефективність, вдосконалені Інтернет-протоколи та управління безпекою.

2 МЕТОДИ ЗАХИСТУ ПРИСТРОЇВ В ІОТ

2.1. Загальні шари IoT та модель злиття даних

Загальна архітектура моделі IoT, як показано на рисунку 2.1, складається з шарів пристрою, хмари та кінцевого користувача. Рівень пристрою складається з пулу бездротових сенсорних пристроїв з підтримкою Інтернету, схеми збору даних та протоколів зв'язку для передачі даних у локальне або віддалене сховище для подальшої обробки.

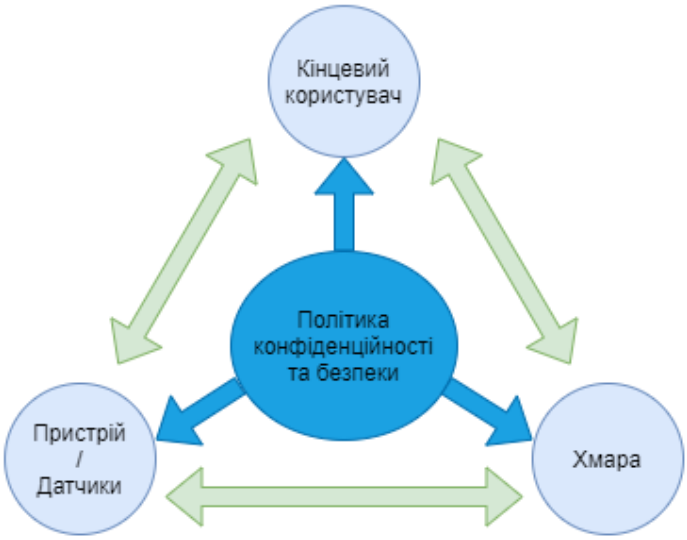

Рисунок 2.1 – Загальна модель Інтернету речей (IoT) із політикою конфіденційності та безпеки

Ці пристрої дозволяють користувачеві збирати дані в режимі реального часу з різними частотами збору. У хмарному шарі зберігаються дані, зібрані з датчиків, для подальшої обробки, видалення шуму, вилучення функцій та масажу даних. Пізніше ці дані подаються в систему підтримки прийняття рішень, яка здійснює комплексний аналіз даних та штучний інтелект для прийняття рішення щодо стану здоров’я людини. Кінцевий рівень

користувача, що складається з користувача-одержувача, може мати різні форми. Викликають занепокоєння розумні пристрої, де існують проблеми безпеки та конфіденційності. У межах цих трьох рівнів додано перелік підшарів або модулів, що забезпечують надійність системи підтримки прийняття рішень. Для того, щоб забезпечити оперативну передачу та обробку даних, щоб надати критичне рішення, яке не може чекати, поки дані будуть відправлені в хмару, я представляю можливість обчислювальних технологій, яка може прийняти таке розумне рішення, і одночасно зберегти копію даних і надіслати його на хмарний шар для обробки та тривалого зберігання [11].

У деяких випадках нам потрібно надсилати команди чи інструкції на деякі носні пристрої, щоб оновити їх швидкість або функціональність, і для цього знадобляться інші протоколи та процедури безпеки.

Рисунок 2.2 – Розтягнутий варіант загальної моделі Інтернету речей (IoT)

На рисунку 2.2 показано розтягнутий варіант загальної моделі. Бачимо додавання нових шарів; край і туман. Обидва рівні можуть подолати проблеми затримки завдяки використанню служб хмарного шару і можуть швидше приймати рішення. Обчислення краю здійснюється на пристроях, до яких датчики прикріплені або фізично закриті. Вони надають рішення та

контроль у реальному часі джерелам даних, і в той же час спілкуються з іншими рівнями для передачі даних для злиття, зберігання та аналітики. Туманний обчислювальний шар переміщує крайні обчислювальні роботи до більш потужних обчислювальних ресурсів, які підключені до локальної мережі та фізично віддаленіші від датчиків та джерел даних. Ці додаткові переваги створюють більше проблем із безпекою та конфіденційністю.

2.2 Політика безпеки та конфіденційності.

Безпека та конфіденційність в Інтернеті речей (IoT) залишається серйозною проблемою через неоднорідну природу великомасштабних пристроїв та її вразливість в робочому середовищі. Але, з іншого боку, програми пристроїв IoT охоплюють різні домени від менших масштабів до більших, таких як від розумного Gird до розумного міста. Однак ця популярність пристроїв IoT обмежена кібератаками та загрозами безпеці. Згідно з аналізом, кілька поширених пристроїв IoT мають в середньому 25% вразливостей на кожен пристрій. Ця тенденція в Інтернеті речей призвела до серйозних рішень щодо безпеки. Пристрої IoT страждають від обчислювальної обробки, малої потужності та обмеженої пам'яті. Система IoT складається з трьох компонентів, таких як сенсорний блок, що має велику кількість датчиків, виконавчих механізмів та мобільних терміналів для виявлення фізичного середовища. Ця тендітна та проста структура IoT робить її більш вразливою до загроз, пов'язаних з безпекою IoT.

Кожен рівень моделі IoT створює виклики безпеці та одночасно можливість застосовувати стандарти та протоколи безпеки та конфіденційності. Безпека мікропрограми та апаратне забезпечення перевіряють автентичність, проте це відбувається за рахунок енергоспоживання, оскільки деякі бездротові пристрої, такі як носимі пристрої, працюють від батареї.

Такі заходи безпеки повинні бути переглянуті для досягнення обмежень як у сфері безпеки, так і в потужності. На хмарному рівні заходи безпеки повинні забезпечувати мережевий протокол між крайовим та туманним вузлами, а іноді і датчиками. Протокол передачі повідомлень, шифрування точка-точка та сертифікати забезпечують менше шпигування та реєстрації даних. На рівні обробки даних та рівня кінцевого користувача нам потрібно забезпечити, щоб довготривале зберігання даних та обробка даних у реальному часі були захищені від SQL-ін'єкцій, обнюхування та фішингових атак сценаріїв, якщо сервісний сертифікат оновлений та відповідає вимогам Стандартів HIPPA (в системах охорони здоров'я). Злиття даних може надати іншому доступу хакерам для ідентифікації користувача, отже, порушення конфіденційності. Оскільки пристрої IoT можуть приєднуватися і виходити з мережі датчиків та джерел даних, це додає більшої складності стандартним методам заходів безпеки, отже, необхідність у нових інтелектуальних та адаптивних заходах безпеки.

2.2.1 Безпека

IoT відрізняється від традиційних комп'ютерів та обчислювальних пристроїв, робить його більш вразливим до проблем безпеки різними способами:

- багато пристроїв в Інтернеті речей призначені для масового розгортання. Чудовим прикладом цього є датчики;
- зазвичай розгортання IoT складається з набору однакових або майже однакових приладів, що мають подібні характеристики. Ця схожість посилює масштаб будь-якої вразливості в системі безпеки, яка може суттєво вплинути на багато з них;
- подібним чином багато установ розробили посібники для проведення оцінки ризиків. Цей крок означає, що ймовірна кількість посилань, взаємопов'язаних між пристроями IoT, є безпрецедентною. Зрозуміло також,

що багато з цих пристроїв можуть встановлювати зв’язки та взаємодіяти з іншими пристроями автоматично нерегулярно. Вони вимагають розгляду доступних інструментів, прийомів і тактик, які пов'язані з безпекою IoT.

Незважаючи на те, що проблема безпеки [12] в інформаційно-технологічному секторі не нова, впровадження IoT представляє унікальні проблеми, які потрібно вирішити. Від споживачів вимагається довіряти пристроям Інтернету речей, а послуги дуже захищені від слабких місць, особливо, оскільки ця технологія продовжує ставати все більш пасивною та вбудованою у наше повсякденне життя. Зі слабо захищеними приладами та послугами IoT це один із дуже важливих шляхів, що використовуються для кібератак, а також викриття даних користувачів, залишаючи потоки даних незахищеними належним чином.

Характер взаємозв'язку пристроїв IoT означає, що якщо пристрій погано захищений і підключений, він може мати вплив на безпеку та стійкість в Інтернеті на міжнародному рівні. Таку поведінку просто спричиняє виклик широкого використання однорідних пристроїв IoT. Окрім можливості деяких пристроїв мати можливість механічного зв’язку з іншими пристроями, це означає, що всі користувачі та розробники IoT зобов'язані забезпечити, щоб вони не піддавали інших користувачів, а також сам Інтернет потенційній шкоді. В даний час в Світовому центрі спостерігається спільний підхід, необхідний для розробки ефективного та відповідного вирішення проблем IoT.

Наприклад, коли йдеться про автентифікацію, IoT стикається з різними вразливими місцями, які залишаються однією з найважливіших проблем забезпечення безпеки багатьох програм. Використовувана автентифікація обмежена тим, як вона захищає лише одну загрозу, наприклад, відмову в обслуговуванні (DoS) або атаки відтворення. Інформаційна безпека є однією з найважливіших вразливих сфер автентифікації IoT через поширеність програм, які є ризикованими через їх природну багатоманітність у зборі даних в середовищі IoT. Якщо ми можемо, наприклад, візьмемо приклад

безконтактних кредитних карток. Ці картки дозволяють читати номери карток та імена без аутентифікації IoT; це дає можливість хакерам купувати товари, використовуючи номер банківського рахунку власника картки та їх особу.

Однією з найбільш розповсюджених атак в Інтернеті речей є людина посередині, де сторонній канал зв'язку з викраденням прагне підробляти ідентифікаційні дані прощупуваних вузлів, які беруть участь в обміні мережею. Людина посередині атаки ефективно змушує банківський сервер визнати транзакцію здійсненою як дійсну подію, оскільки суперник не повинен знати особу передбачуваної жертви.

2.2.2 Конфіденційність

Перспектива корисності IoT залежить від того, наскільки добре вона може поважати вибір приватного життя людей. Побоювання щодо конфіденційності та потенційної шкоди, що виникають разом з IoT, можуть бути значними для стримування повного прийняття IoT. Важливо знати, що права на конфіденційність та дотримання конфіденційності користувачів є фундаментальними для забезпечення впевненості та впевненості користувачів у Інтернеті речей, підключеному пристрої та супутніх послугах. Проводиться багато роботи для того, щоб IoT переосмислив питання конфіденційності, такі як посилення спостереження та відстеження. Причина занепокоєння щодо конфіденційності полягає у всюдисущих інтегрованих артефактах розвідки, де процес відбору проб та розповсюдження інформації в Інтернеті речей може здійснюватися майже в будь-якому місці. Всюдисущий зв’язок через доступ до Інтернету також є важливим фактором, який допомагає зрозуміти цю проблему, тому що, якщо не буде створений унікальний механізм, тоді буде набагато зручніше отримувати доступ до особистої інформації з будь-якого куточка світу.

2.3 Заходи безпеки IoT.

Загальні заходи безпеки IoT включають:

- включення безпеки на етапі проектування. Розробники IoT повинні включати безпеку на початку розробки будь-яких споживчих, корпоративних або промислових пристроїв. Увімкнення безпеки за замовчуванням є критично важливим, а також забезпечення найновіших операційних систем та використання захищеного обладнання;

- швидко закодовані облікові дані ніколи не повинні бути частиною процесу проектування. Додатковим заходом, який можуть вжити розробники, є вимагання оновлення облікових даних користувачем перед функціями пристрою. Якщо пристрій постачається із типовими обліковими даними, користувачі повинні оновити їх, використовуючи надійний пароль, багатофакторну автентифікацію або біометричні дані, де це можливо;

- PKI та цифрові сертифікати. Інфраструктура відкритих ключів (PKI) та 509 цифрових сертифікатів відіграють важливу роль у розробці захищених пристроїв IoT, забезпечуючи довіру та контроль, необхідні для розповсюдження та ідентифікації відкритих ключів шифрування, безпечного обміну даними через мережі та перевірки ідентичності;

- безпека API. Захист індикатора продуктивності додатків (API) є надзвичайно важливим для захисту цілісності даних, що надсилаються з пристроїв IoT до внутрішніх систем, а також для забезпечення взаємодії з API лише авторизованих пристроїв, розробників та програм;

- управління особистістю. Надання кожному пристрою унікального ідентифікатора є критично важливим для розуміння того, що таке пристрій, як він поводиться, інших пристроїв, з якими він взаємодіє, та належних заходів безпеки, які слід вжити для цього пристрою;

- апаратна безпека. Затвердіння кінцевої точки включає в себе виготовлення пристроїв, захищених від фальсифікації або захисту від фальсифікацій. Це особливо важливо, коли пристрої будуть

використовуватися в суворих умовах або там, де вони не будуть фізично контролюватися;

- сильне шифрування є критичним для забезпечення зв'язку між пристроями. Дані у стані спокою та під час транзиту слід захищати за допомогою криптографічних алгоритмів. Сюди входить використання ключового управління життєвим циклом;

- безпека мережі. Захист мережі IoT включає забезпечення безпеки портів, відключення переадресації портів і ніколи не відкривання портів, коли це не потрібно; використання антивірусного програмного забезпечення, брандмауерів та системи виявлення вторгнень/системи запобігання вторгненню; блокування несанкціонованих IP-адрес; та забезпечення того, щоб системи були виправлені та оновлені;

- контроль доступу до мережі. NAC може допомогти визначити та скласти інвентаризацію пристроїв IoT, що підключаються до мережі. Це забезпечить базову лінію для пристроїв відстеження та моніторингу. Пристрої IoT, яким потрібно підключатися безпосередньо до Інтернету, слід сегментувати у свої власні мережі та мати обмежений доступ до корпоративної мережі. Сегменти мережі повинні контролювати аномальну активність, де можна вжити заходів у разі виявлення проблеми;

- шлюзи безпеки. Діючи як посередник між пристроями IoT та мережею, шлюзи безпеки мають більше обчислювальної потужності, пам'яті та можливостей, ніж самі пристрої IoT, що забезпечує їм можливість реалізувати такі функції, як брандмауери, щоб гарантувати, що хакери не можуть отримати доступ до пристроїв IoT, які вони підключають;

- управління виправленнями/постійне оновлення програмного забезпечення. Забезпечення засобів оновлення пристроїв та програмного забезпечення через мережеві підключення або за допомогою автоматизації є критично важливим. Координоване розкриття вразливостей також важливо для якнайшвидшого оновлення пристроїв. Розгляньте також стратегії закінчення життя;

- IoT та безпека операційної системи є новими для багатьох існуючих команд безпеки. Дуже важливо постійно інформувати працівників служби безпеки про нові чи невідомі системи, вивчати нові архітектури та мови програмування та бути готовими до нових викликів безпеці. Команди на рівні С та з питань кібербезпеки повинні регулярно проходити навчання, щоб не відставати від сучасних загроз та заходів безпеки;

- інтеграція команд. Поряд із навчанням може бути корисним об'єднання різнорідних та регулярно відключених команд. Наприклад, співпраця розробників програм із спеціалістами з безпеки може допомогти забезпечити додавання належних елементів керування до пристроїв на етапі розробки;

- освіта споживачів. Споживачі повинні бути поінформовані про небезпеку систем IoT та надані їм кроки для забезпечення безпеки, такі як оновлення облікових даних за замовчуванням та застосування оновлень програмного забезпечення. Споживачі також можуть зіграти певну роль у вимаганні від виробників пристроїв створення захищених пристроїв та відмові від використання тих, які не відповідають високим стандартам безпеки.

При будь-якому розгортанні IoT критично важливо зважити вартість безпеки та ризики до впровадження.

2.4 Управління безпекою Інтернету речей

Величезний обсяг пристроїв Інтернету речей робить їх безпеку першочерговим завданням і має вирішальне значення для майбутнього добробуту екосистеми Інтернету.

Для користувачів пристроїв це означає дотримання основних найкращих практик безпеки, таких як зміна паролів безпеки за замовчуванням та блокування непотрібного віддаленого доступу (наприклад, коли це не потрібно для функціонування пристрою).

Постачальники та виробники пристроїв, навпаки, повинні застосовувати більш широкий підхід і інвестувати значні кошти в забезпечення інструментів управління IoT. Кроки, які слід вжити, включають:

- попередньо повідомляти користувачів про пристрої, на яких встановлено застарілі версії програмного забезпечення / ОС;
- застосування розумного управління паролями (наприклад, обов'язкові зміни пароля за замовчуванням);
- вимкнення віддаленого доступу до пристрою, якщо це не потрібно для основних функцій;
- представляти жорстку політику контролю доступу для API;
- захист центрів C&C від спроб компрометації та DDoS-атак;
- хмарний WAF Imperva допомагає виробникам IoT захищати свої центри C&C, надаючи сучасні послуги фільтрації трафіку, які забезпечують доступ до їх API лише авторизованим та автентифікованим клієнтським запитам;
- для додаткової надійності також є функція балансування навантаження та відмов, які допомагають операторам долати органічні стрибки трафіку, такі як ті, що можуть виникнути після випуску нового виправлення мікропрограми.

2.5 Сертифікація IoT-пристроїв

Фахівці визнають, що причина виникнення проблем безпеки інтернету речей полягає зовсім не в недостатній кваліфікації розробників, а в гонитві за прибутком. Для компаній важливо прискорити випуск нового пристрою на ринок. Деякі виробники вважають за краще пожертвувати захищеністю, заради отримання переваги перед конкурентами. Багато компаній і сьогодні випускають розумні гаджети, не вкладаючи великі ресурси грошей і часу в тестування кодів, доопрацювання систем безпеки. З цієї причини ринок росте

дуже швидко, технології розвиваються, але страждають користувачі. Змусити виробників переглянути своє ставлення до безпеки виготовляються «розумних» гаджетів може введення сертифікації. Це не революційна ідея, проте в перспективі вона дає можливість зменшити масштаби проблеми.

В ідеалі, сертифікація повинна бути досить простою і швидкою для виробника, щоб не стати перепоною на шляху прогресу, але в той же час вона повинна забезпечувати користувачам хороший захист від будь-яких можливих атак.

В даний час в області сертифікації розумних девайсів працює кілька приватних організацій, наприклад, Online Trust Alliance (OTA), яка підготувала ініціативу для вирішення проблеми. Так, був випущений унікальний список критеріїв для розробників нового обладнання, дотримання яких дозволяє підвищити безпеку і захистити конфіденційні дані користувачів.

Сертифікація підтверджує, що пристрій або система забезпечують необхідний рівень безпеки з урахуванням можливих ризиків. Також вона виступає підтвердженням, що нові версії програмного забезпечення для девайсів НЕ будуть приводити до втрати безпеки.

Однак сертифікація не може гарантувати захищеність на сто відсотків, це лише один з рівнів захисту. І наявність такого документа все ж залишає ймовірність отримання зловмисниками доступу до пристрою.

2.6 Захист програмного коду IoT

При включенні кожен пристрій завантажується і запускає певний виконуваний код. Важливо бути впевненими в тому, що пристрої будуть робити тільки те, на що їх запрограмували, а сторонні не зможуть перепрограмувати на зловмисну поведінку. Тобто, першим кроком у захисті пристроїв є захист коду, щоб гарантовано завантажувався і запускався тільки потрібний нам код. На щастя, багато виробників вже вбудували можливості

безпечного завантаження в свої чіпи. Схожим чином справи йдуть і з високорівневим кодом – різні перевірені часом клієнтські бібліотеки з відкритим вихідним кодом, на кшталт OpenSSL, можуть використовуватися для перевірки підпису і дозволу коду тільки з авторизованого джерела.

Внаслідок цього все більшого поширення набувають підписані прошивки, образи завантаження і більш високорівневий вбудований код, в тому числі підписані базові програмні компоненти, куди входять будь-які операційні системи. Все частіше зустрічаються не просто підписані прикладні програми, а взагалі весь код на пристрої.

Такий підхід гарантує, що всі критичні компоненти систем IoT: датчики, механізми, контролери та реле сконфігуровані правильно - на запуск тільки підписаного коду і ніколи не запустять непідписаний код. Доброю манерою було б дотримуватися принципу «ніколи не довіряти не підписаному коду». Логічним продовженням було б «ніколи не довіряти не підписаним даним і, тим більше, не підписаним конфігураційним даним».

Використання сучасних засобів перевірки підпису і поширення апаратної реалізації безпечного завантаження, ставлять серйозне завдання перед багатьма компаніями – управління ключами і контроль доступу до ключів для підпису коду і захисту програмно-апаратних засобів. На щастя, деякі центри сертифікації пропонують хмарні сервіси, які роблять простіше, безпечніше і надійніше адміністрування програм для підписування коду і гарантують суворий контроль, хто може підписувати код, відкликати підписи, і як ключі для підписання і відкликання захищені.

Виникають ситуації, коли програмне забезпечення потрібно оновити, наприклад, в цілях безпеки, але при цьому необхідно врахувати вплив оновлень на заряд батареї. Операції перезапису даних збільшують споживання енергії і скорочують період автономної роботи пристрою.

З'являється необхідність підписати і оновити окремі блоки або фрагменти таких оновлень, а не монолітні образи цілком або бінарні файли. Тоді програмне забезпечення, підписана на рівні блоків або фрагментів,

можна оновлювати з набагато більш тонкої деталізацією, не жертвуючи безпекою або зарядом батареї. Для цього не потрібна обов'язково апаратна підтримка, таку гнучкість можна досягти від передзавантажувального середовища, яка може працювати на безлічі embedded-пристроїв.

Якщо час автономної роботи настільки важливий, чому б просто не конфігурувати пристрій з незмінної прошивкою, яку ніхто не може змінити або оновити? На жаль, ми змушені припустити, що пристрої в польових умовах схильні до реверс-інжинірингу для шкідливих цілей. Після його проведення виявляються і експлуатуються уразливості, які необхідно патчити якомога швидше.

Обфускація і шифрування коду можуть істотно уповільнити процес реверс-інжинірингу та відбити бажання продовжувати атакувати у більшості зловмисників. Але ворожі спецслужби або міжнаціональні деструктивні організації все-таки здатні це зробити навіть для програм, захищених за допомогою обфускації і шифрування, перш за все тому, код повинен бути дешифрований для запуску. Такі організації знайдуть і скористаються уразливими, які не були вчасно пропатчити.

У зв'язку з цим можливості віддаленого поновлення (OTA) мають вирішальне значення і повинні бути вбудовані в пристрої до того, як вони покинуть завод. OTA-оновлення software і firmware дуже важливі для підтримки високого рівня захищеності пристрою. Детальніше цей момент ми з вами ще розглянемо в розділі «Контроль пристроїв». Проте, обфускація, сегментоване підписання коду і OTA-оновлення в кінцевому рахунку повинні бути щільно об'єднані між собою для ефективної роботи.

До речі, і сегментоване, і монолітне підписання коду використовують модель довіри на основі сертифікатів, описану в попередньому розділі «Безпека зв'язку», а використання ECC при підписанні коду може забезпечити ті ж самі переваги високого рівня безпеки в поєднанні з високою продуктивністю і низьким енергоспоживанням. У цій ситуації пропонуються

наступні рекомендації по довжині ключа для підпису коду IoT, де безпека має значення:

- мінімум 224-bit ECC для сертифікатів кінцевих об'єктів з переважним 256-bit і 384-bit;

- мінімум 521-bit ECC для кореневих сертифікатів, оскільки, як правило, очікується, що підписаний код буде використовуватися роками або навіть десятиліттями після підписання, а підписи повинні бути досить сильними, щоб залишатися надійними протягом такого тривалого часу.

2.7 Ефективний хостовий захист для IoT

Після захисту зв'язку і реалізації безпечного завантаження добре керованого пристрою, необхідний захист на етапі експлуатації. Хостовий захист вирішує цю задачу. IoT-пристрої стикаються з багатьма загрозами, в тому числі шкідливим кодом, який може поширюватися через перевірені з'єднання, скориставшись уразливими або помилками в конфігурації. В таких атаках часто експлуатуються кілька слабких місць, включаючи, але не обмежуючись:

- невикористання перевірки підпису коду і безпечну завантаження;
- погано реалізовані моделі перевірки, які можна обійти.

Атакуючі часто використовують ці недоліки для установки бекдорів, сніфферів, програмного забезпечення для збору даних, можливості передачі файлів для витягання конфіденційної інформації з системи, а іноді навіть для інфраструктури command & control (C & C) для маніпулювання поведінкою системи. Особливо тривожить здатність деяких зловмисників експлуатувати вразливості для установки шкідливих програм прямо в пам'ять вже працюють систем IoT. Причому іноді вибирається такий спосіб зараження, при якому шкідлива програма зникає після перезавантаження пристрою, але встигає наносити величезної шкоди. Це працює, тому що деякі системи IoT і багато промислових системи майже ніколи не перезавантажуються.

Для відділу безпеки в цьому випадку ускладнюється можливість виявлення використаної уразливості в системі і розслідування походження атаки. Іноді такі атаки відбуваються через IT-мережу, підключену до промислової мережі або до мережі IoT, в інших випадках атака відбувається через інтернет або через прямий фізичний доступ до пристрою. Не важливо, який був вихідний вектор інфекції, але якщо він не виявлений, то перше скомпрометований пристрій як і раніше залишається довіреною і стає провідником для зараження іншої мережі, будь то автомобільна мережа транспортного засобу або ціла виробнича мережа заводу.

Безпека IoT повинна бути комплексною. Закриваючи вікна, залишати двері відчиненими – неприйнятно. Всі вектори загроз повинні придушуватися. На щастя, в поєднанні з надійним підписом коду і моделлю перевірки, хостовий захист може допомогти захистити пристрій від безлічі небезпек. У хостовому захисту використовується ряд технологій захисту, в тому числі харденінг, розмежування доступу до системних ресурсів, пісочниця, захист на основі репутації і поведінки, захист від шкідливих програм і, нарешті, шифрування.

Залежно від потреб конкретної системи IoT комбінація цих технологій може забезпечити найвищий рівень захисту для кожного пристрою. Харденінг, розмежування доступу до ресурсів і пісочниця захистять всі «двері» в систему. Вони обмежують мережеві підключення до додатків і регламентують вхідний і вихідний потік трафіку, захищають від різних експлойтів, переповнення буфера, цілеспрямованих атак, регулюють поведінку додатків, при цьому дозволяють зберегти контроль над пристроєм. Такі рішення ще можуть використовуватися для запобігання несанкціонованого використання знімних носіїв, блокування конфігурації та налаштувань пристрою і навіть для деескалації користувальницьких привілеїв, якщо потрібно.

Хостовий захист має можливості аудиту і оповіщення, допомагаючи відстежувати журнали і події безпеки. Технології на основі політик можуть

працювати навіть в середовищах без підключення до інформаційної мережі або при обмеженій обчислювальної потужності, необхідної для використання традиційних технологій.

Технологія захисту на основі репутації може використовуватися для визначення сутності файлів по їхньому віку, поширеності, розташування і до решти для виявлення небезпек, що не виявляються іншими засобами, а також давати уявлення про те, чи слід довіряти новому пристрою навіть при успішній перевірці автентичності. Таким способом можна ідентифікувати загрози, які використовують мутуючий код або адаптують свою схему шифрування, просто відокремлюючи файли з високим ризиком від безпечних, швидко і точно виявляючи шкідливі програми, незважаючи на всі їхні хитрощі.

Поєднання застосовуваних технологій буде залежати від конкретної ситуації, але наведені вище можуть об'єднуватися для захисту пристроїв, навіть в середовищах з обмеженими обчислювальними ресурсами.

2.8 Загрози та напади на рівень елементів

Елементний рівень складається з різних вузлів і датчиків для збору даних із підключеного мережевого середовища. Вузли та датчики піддаються різним загрозам, таким як:

- несанкціонований доступ. Шар Element використовував вузли та датчики, такі як RFID, мітки, мітки штрих-коду, виконавчі механізми та інтелектуальні пристрої виявлення для збору даних із навколишнього середовища; через відсутність служб автентифікації, несанкціоновані сторони можуть отримати доступ до даних та змінити їх, або навіть видалити дані;

- підслуховування. Інформація, зібрана бездротовими компонентами, такими як RFID та теги, легко читається зловмисниками, як уже згадувалося в посиланні. Зловмисники можуть використовувати дані для злому будь-якої

системи IoT або винюхування такої важливої інформації, як пароль або конфіденційна інформація користувачів;

- підробка. Спуфінг – це зловмисники, які надсилають фальшиву інформацію на вузли, і датчики роблять вигляд, що вони діють як оригінальна несправність, тоді зловмисники можуть мати повний доступ до системи.

2.8.1 Захист шару елемента

Елементний шар – це найнижчий шар із чотирьох шарів системного середовища IoT. Елементний шар складається з датчиків і вузлів, ці пристрої піддаються таким загрозам, як:

- послуги захисту рівня елемента. Служби автентифікації захищають рівень елементів від атак несанкціонованого доступу. Служби контролю доступу можуть захистити елементний рівень від атак підслуховування. Служби конфіденційності потрібні для захисту рівня елемента від атак підміни. Таким чином, служби автентифікації, контролю доступу та конфіденційності захищають рівень елемента від таких атак, як несанкціонований доступ, прослуховування та підробка;
- механізми захисту рівня елементів. Служби автентифікації на рівні елементів використовують хеш-алгоритми для забезпечення цифрового підпису для протидії атакам несанкціонованого доступу, механізм таблиці контролю доступу протидіє атакам підслуховування та інфраструктура відкритого ключа (PKI) забезпечує конфіденційність даних, зібраних датчиками та смарт-пристроїв.

2.9 Загрози та атаки на мережевий рівень

Мережевий рівень передає дані, які зібрані вузлами та датчиками, на термінал, для передачі даних використовувалася бездротова сенсорна

мережа, на мережевому рівні є кілька проблем безпеки, таких як:

- відмова в обслуговуванні (DoS). DoS-атака – це випадки, коли зловмисники надсилають багато марних даних, щоб зробити мережевий трафік затопленим. Через величезне споживання системних ресурсів система IoT буде заблокована для доступу авторизованих користувачів;

- атака "Людина посередині". Ця атака є своєрідним підслуховуванням того, що несанкціоновані зловмисники можуть контролювати спілкування між двома сторонами. Зловмисник може отримати корисну інформацію за каналами зв'язку;

- введення зловмисного коду. У цьому випадку зловмисник компрометує вразливі вузли та датчики, вводячи шкідливі коди, і атакує систему IoT. Результат може призвести до вимкнення мережі, і зловмисники можуть отримати контроль над системою.

2.9.1 Безпека мережевого рівня

Мережевий рівень передає дані, зібрані вузлами та датчиками, на верхній рівень, який є сервісним рівнем. На мережевому рівні слід вирішити декілька проблем безпеки, таких як:

- служби безпеки мережевого рівня. Доступність та відсутність відмови в обслуговуванні захищають мережевий рівень від атак відмови в обслуговуванні, служба цілісності та захисту захищає мережевий рівень від атак "людина посеред", антивірусні служби захищають мережевий рівень від шкідливих атаки введення коду;

- механізми захисту мережевого рівня. Доступність та відмова в послугах використовуються через фільтрацію маршрутизатора для протидії атакам відмови в обслуговуванні мережевого рівня. Шифрування даних, необхідне для протидії атакам "людина посередині", та антивірусний механізм безпеки, необхідний для протидії атакам введення шкідливого коду.

2.10 Загрози та атаки на рівень обслуговування

Сервісний рівень обробляє дані та забезпечує посилання на сховище для зібраних даних з рівня елементів. Захист сервісного рівня повинен запобігати таким атакам, як:

- DoS-атака. DoS-атака на сервісному рівні схожа на мережевий рівень, оскільки зловмисники надсилають багато марних даних, щоб зробити мережевий трафік затопленим, таким чином, величезне споживання системних ресурсів вичерпує систему IoT і спричиняє неможливість доступу користувачів до системи;

- несанкціонований доступ. Несанкціоновані зловмисники могли отримати доступ до сервісного рівня, який забезпечує інтерфейс до даних та служб зберігання; таким чином, зловмисники можуть модифікувати або видалити важливі дані та спричинити фатальні проблеми для системи IoT;

- зловмисний інсайдер. Зловмисна інсайдерська атака відбувається зсередини середовища IoT, яке використовує дані для особистого використання. До цих даних дуже легко отримати доступ зсередини, і це можуть зробити лише авторизовані користувачі. Ця загроза відрізняється від несанкціонованого доступу і вимагає інших механізмів протидії загрозі.

2.10.1 Захист рівня обслуговування

Сервісний рівень обробляє дані та надає посилання на сховище для зібраних даних з рівня елементів. Сервісний рівень займається проблемами безпеки, такими як:

- служби безпеки рівня обслуговування. Доступність і відсутність відмови в обслуговуванні захищають рівень обслуговування від атак відмови в обслуговуванні. Служба контролю доступу та авторизації захищає рівень обслуговування від несанкціонованих атак доступу, а служба журналу аудиту ми можемо захистити рівень послуг від зловмисних інсайдерських атак;

- механізми захисту рівня обслуговування. Доступність та відмова в обслуговуванні за допомогою системи виявлення вторгнень (IDS) може протидіяти атакам відмови в обслуговуванні. Механізм контролю доступу використовується для протидії несанкціонованим атакам доступу, а моніторинг подій необхідний для протидії зловмисним інсайдерським атакам.

2.11 Загрози та атаки на рівень додатків

Прикладний рівень складається з різноманітних програм IoT. Потрібно вирішити такі проблеми безпеки, як:

- DDoS-атака. Розподілена атака відмови в послугах (DDoS) на рівні додатків є сучасною. Що стосується нешифрованих пристроїв, зловмисник може легко порушити систему та спричинити проблеми із конфіденційністю даних для користувачів. Жертви не мають доступу до послуг системи і навряд чи помітили, що DDoS-атаки відбувалися в системі IoT. DDoS-атаки відрізняються від DoS-атак. Атаки розподіленого відмови в послугах (DDoS) запускаються з безлічі різних підключених пристроїв; ці підключені пристрої розподіляються в середовищі IoT;

- введення зловмисного коду – це випадки, коли зловмисники зламують систему та вводять певний зловмисний код, щоб отримати доступ адміністрації та контролю системи IoT. Зловмисники можуть отримати конфіденційні дані або видалити важливі дані системи. Ця атака на прикладний рівень вимагає іншого механізму, ніж коли зловмисні атаки відбуваються на нижчих рівнях;

- фішингова атака – це свого роду атака електронною поштою; авторизовані користувачі заманили відкрити електронне повідомлення, і зловмисник зламує систему, щоб отримати контроль доступу до системи IoT. Зловмисники можуть отримати конфіденційні повідомлення або конфіденційні дані, щоб отримати контроль над цілою системою.

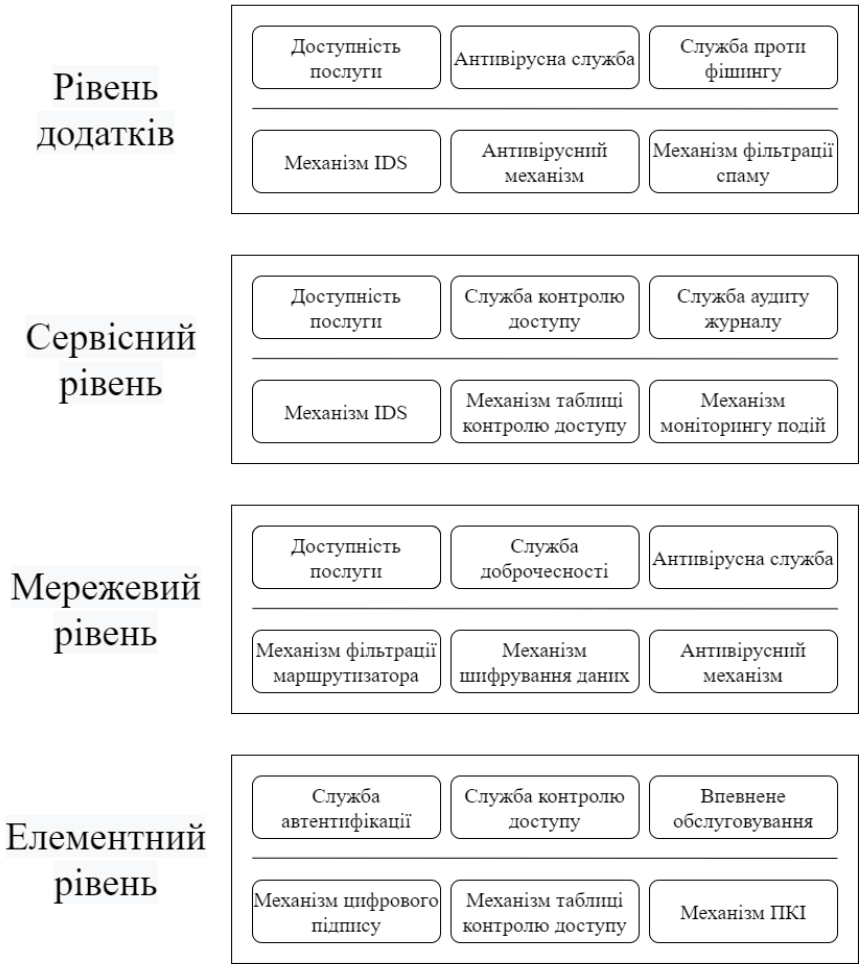

Рисунок 2.3 – Послуги та механізми безпеки на кожному рівні.

На цьому рівні може статися кілька порушень безпеки, таких як:

- служби захисту рівня додатків. Доступність та відсутність відмови в обслуговуванні захищають рівень додатків від розподілених атак відмови в обслуговуванні. Антивірусні служби захищають рівень додатків від зловмисних атак введення коду, а антифішинг захищають рівень додатків від фішингових атак;

- механізми захисту рівня додатків. Наявність і не відмова в обслуговуванні за допомогою IDS необхідні на цьому рівні для протидії розподіленим атакам відмови в обслуговуванні. Антивірусний механізм необхідний для протидії атакам введення зловмисного коду, а механізм фільтрації спаму може протидіяти фішинговим атакам. У цьому розділі ми

використовували концепцію служб безпеки та механізмів, визначених у МСЕ-Т (X.800). ITU-T X.800 вітає деякі механізми безпеки для надання послуг безпеки, визначених стандартами.

У таблиці 1.1 та на рисунку 2.3 показані служби безпеки, а також механізми на кожному рівні.

Таблиця 1.1 – Служби та механізми безпеки

Шари	Напади загрози.	Служби безпеки.	Механізми безпеки
Елемент	Несанкціонований доступ Підслуховування Підробка	Аутентифікація Управління доступом Конфіденційність	Цифровий підпис Таблиця контролю доступу PKI
Мережа	Відмова в наданні послуг Людина посередині Введення зловмисного коду	Доступність Чесність Антивірус	Фільтрування маршрутизатора Шифрування даних Антивірус
Обслуго-вування	Відмова в наданні послуг Несанкціонований доступ Шкідливий інсайдер	Доступність Управління доступом Журнал аудиту	IDS Таблиця контролю доступу Моніторинг подій
Застосу-вання	DDoS Введення зловмисного коду Фішинг	Доступність Антивірус Захист від фішингу	IDS Антивірус Фільтрування спаму

PKI: Фреймворк забезпечує конфіденційність даних у зв'язку за допомогою шифрування та автентифікації.

IDS: система безпеки для моніторингу трафіку мережі для виявлення DoS-атак.

2.12 Стандарти та протоколи для IoT на кожному рівні

Завдяки існуванню різноманітних мереж, пристроїв та додатків в середовищі IoT використовується низка стандартів, і різні організації беруть участь. Це додає складності дизайну, а також впровадження надійної мережі IoT.

Рисунок 2.4 показує організації та різні протоколи, які сьогодні використовуються для IoT на кожному рівні.

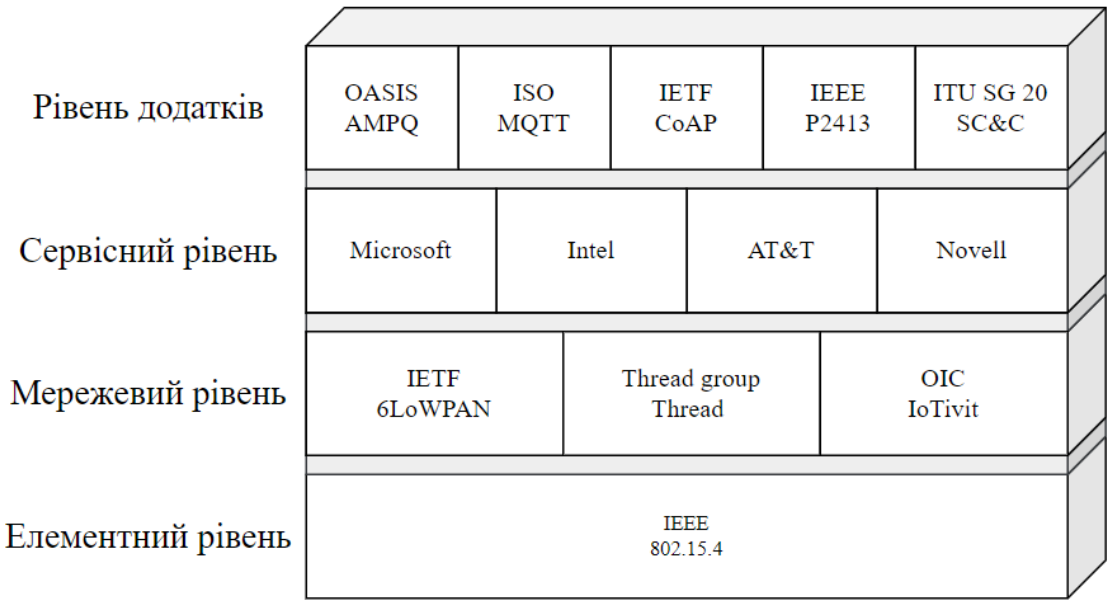

Рисунок 2.4 – Організація та протоколи IoT на кожному рівні

IEEE 802.15.4 – це стандарт, який визначає фізичний рівень (PHY) та зв'язок управління доступом до засобів масової інформації (MAC) для низькошвидкісного недорогого зв'язку між пристроями в бездротових

персональних мережах. IEEE 802.15.4 реалізує вдосконалений стандарт шифрування (AES) симетричний механізм криптографії та підтримує кілька режимів безпеки; ці режими безпеки надають такі послуги безпеки, як конфіденційність, автентифікація та цілісність [13].

Таблиця 1.2 – Послуги безпеки та режими безпеки IEEE 802.15.4

Служба безпеки	Режим безпеки	Механізм безпеки
Конфіденційність	AES-CTR	Дані шифруються за допомогою AES у режимі лічильника за допомогою 128-бітних ключів
Цілісність аутентифікації	AES-CBC-MAC/MIC-32	Дані шифруються за допомогою AES у режимі ланцюжка блоків cypher з кодом автентифікації повідомлень та кодом цілісності повідомлень у 32/64/128-бітних ключах
	AES-CBC-MAC/MIC-64	
	AES-CBC-MAC/MIC-128	
Цілісність автентифікації конфіденційності	AES-CCM-32	Дані шифруються за допомогою AES у режимі ланцюжка лічильників та блоків шифрування за допомогою коду автентифікації повідомлень та коду цілісності повідомлень у 32/64/128-бітних ключах
	AES-CCM-64	
	AES-CCM-128	

6LoWPAN – це мережевий протокол, який передає пакети IPv6 через малопотужне середовище бездротового зв'язку IEEE802.15.4. 6LoWPAN

реалізує протокол маршрутизації (RPL) для механізмів маршрутизації з низьким енергоспоживанням та мереж з втратами (LLN) і має три режими безпеки. RPL реалізує AES із 128-бітними ключами для MAC та підтримує RSA із SHA-256 для цифрових підписів для забезпечення конфіденційності та цілісності. Режими безпеки описані нижче:

- незахищений: у цьому захищеному режимі RPL надсилає дані без використання будь-якого додаткового захисту, і це режим захисту за замовчуванням у протоколі RPL;

- попередньо встановлена: симетричні ключі передаватимуть вузли, приєднуючись до RPL;

- автентифікований: коли новий пристрій приєднується до мережі, авторитет ключа здійснить автентифікацію та авторизує новий пристрій.

На рисунку 2.5 показано формат заголовка повідомлення CoAP. Заголовок CoAP має 4 байти: поле версії з 2 бітами, тип повідомлення з 2 бітами, довжина маркера з 4 бітами, поле коду з 8 бітами та ідентифікатор повідомлення з 16 бітами. Маркер дозволяв CoAP виконувати збіг запиту та відповідей, параметри визначають формат значення довжини, вказуючи номер опції, слідуючи його довжині та значенню.

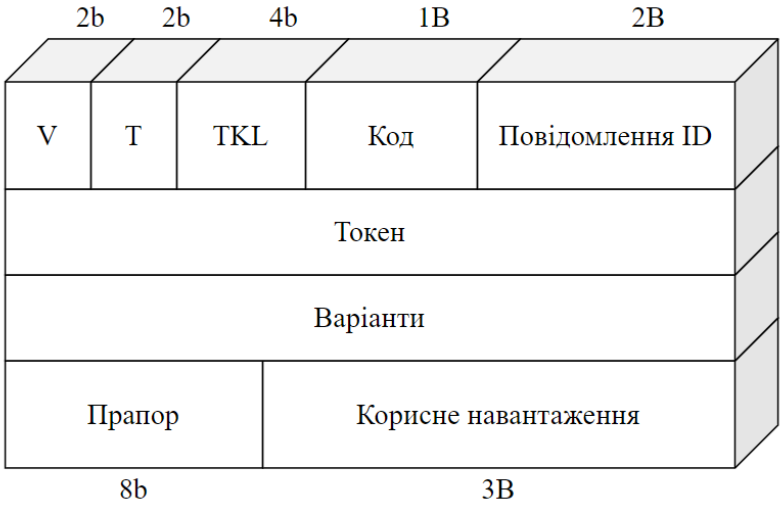

Рисунок 2.5 – Заголовок повідомлення CoAP

CoAP працює над UDP. Прикладний рівень, заснований на UDP, зменшує вимоги до пропускної здатності та підтримує багатоадресну та одноадресну передачу, метою CoAP є пристрої, обмежені ресурсами, такі як мобільний телефон, планшет, ноутбук та пристрої з низьким енергоспоживанням.

Протокол CoAP забезпечує модель зв'язку "запит і відповідь" між кінцевими точками та приймає AES як криптографічний алгоритм для забезпечення таких служб безпеки, як конфіденційність, автентифікація та цілісність.

3 РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ БЕЗПЕКОЮ В ІОТ

3.1 Система управління безпекою IoT

Система управління безпекою IoT (IoTSMS) повинна базуватися на архітектурі мережевої системи IoT. У мережевій системі IoT існує п’ять основних проблем безпеки; кожен з них повинен бути розглянутий до проектування в системі управління безпекою. Ці проблеми безпеки полягають у тому, що розумні датчики легко атакувати, управління безпекою повинно підтримувати малопотужні розумні пристрої, проблеми конфіденційності пристроїв елементного рівня, різні рівні, що стикаються з подібними загрозами, та складність системи та проблеми сумісності.

Ці вимоги означають, що нам потрібно розробити систему управління безпекою для середовища IoT, яка протидіє всім сприйнятим загрозам і сумісна з архітектурою мережі IoT. Іншими словами, оскільки мережеве середовище IoT розроблено як чотиришарова архітектура системи, цілком доречно, щоб управління безпекою мережі було організовано за тими ж принципами, що і шарувата архітектура. Для реалізації цієї концепції ми пропонуємо чотирирівневу систему управління безпекою для середовища IoT, як показано на рисунку 3.1, подібну до тієї, що використовується для функціональної архітектури IPSec.

Запропонована система управління безпекою IoT (IoTSMS) має чотири функціональні рівні. Принципи, які застосовувались для досягнення чотирьох шарів, такі:

- рівень функціональності створюється там, де потрібен інший тип функцій безпеки на різному рівні;
- кожен рівень виконує чітко визначену функцію захисту;
- функціональність кожного рівня вибирається з урахуванням вже існуючих стандартизованих протоколів;

- межі шару вибираються для мінімізації потоку даних через системні інтерфейси;

- кількість шарів сумісна із системними рівнями IoT таким чином, що різні функції безпеки не повинні проходити в одному рівні.

Показана система управління безпекою складається з трьох частин; ліворуч ми показали архітектуру мережевої системи IoT, яка складається з чотирьох шарів. У середній частині знаходиться IoTSMS, який має чотири рівні як управління бізнес-політикою безпеки, управління службами безпеки IoT, управління механізмами безпеки IoT та фундаментальну функцію безпеки IoT. Такі як генератор псевдовипадкових, мультиплікативний зворотний, модульна арифметична функція тощо.

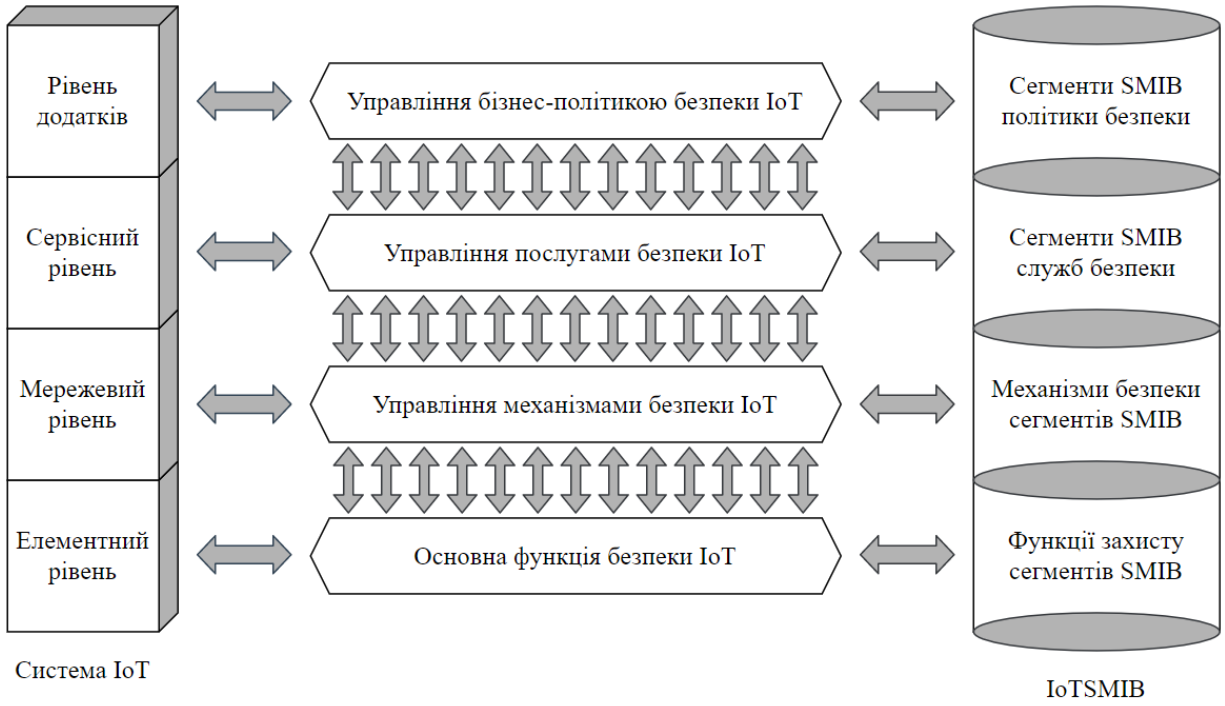

Рисунок 3.1 – Система управління безпекою для IoT

Кожен рівень має відповідну функціональність управління безпекою для забезпечення конфіденційності даних, цілісності даних та доступності даних. Праворуч на цій діаграмі знаходиться інформаційна база управління безпекою IoT (SMIB), SMIB реалізує аутентифікацію рекомендацій X.509

версії 3, серед іншого, для надання концептуальним даним необхідних сегментів ідентифікаторів розумних датчиків, профілів користувачів , список контролю доступу та журнали безпеки [14].

3.2 Функціональні рівні управління безпекою для IoT

Як було зазначено раніше, існує чотири рівні управління безпекою для IoT. Це рівень управління бізнес-політикою безпеки IoT, рівень управління послугами безпеки IoT, рівень управління механізмами безпеки IoT та основний рівень функцій безпеки IoT. Кожен рівень має власну функцію забезпечення захисту системи управління безпекою IoT.

3.2.1 Вимоги до управління бізнес-політикою IoT Security

Рівень управління бізнес-політикою безпеки стосується вимог бізнес-користувачів, таких як запобігання та виявлення всіх атак з різних точок атак, захист конфіденційності всіх інтелектуальних пристроїв та захист системи IoT від атак та попередження збою системи. Рисунок 3.2 показує управління бізнес-політикою безпеки IoT з мінімальними вимогами.

Запобігання атакам	Виявлення атак
Захист конфіденційності	Аварійного відновлення

Рисунок 3.2 – Вимоги до управління бізнес-політикою IoT Security.

3.2.2 Функція служб безпеки IoT

Розділ функціонального рівня служб безпеки IoT надає найпоширеніші послуги безпеки, такі як служба автентифікації, включаючи автентифікацію одноліткової сутності та автентифікацію джерела даних, конфіденційність, мабуть, найпоширеніший аспект безпеки IoT, включаючи конфіденційність з'єднання, конфіденційність без підключення, вибіркову конфіденційність поля та трафік конфіденційність потоку. Інформація в середовищі IoT постійно змінюється.

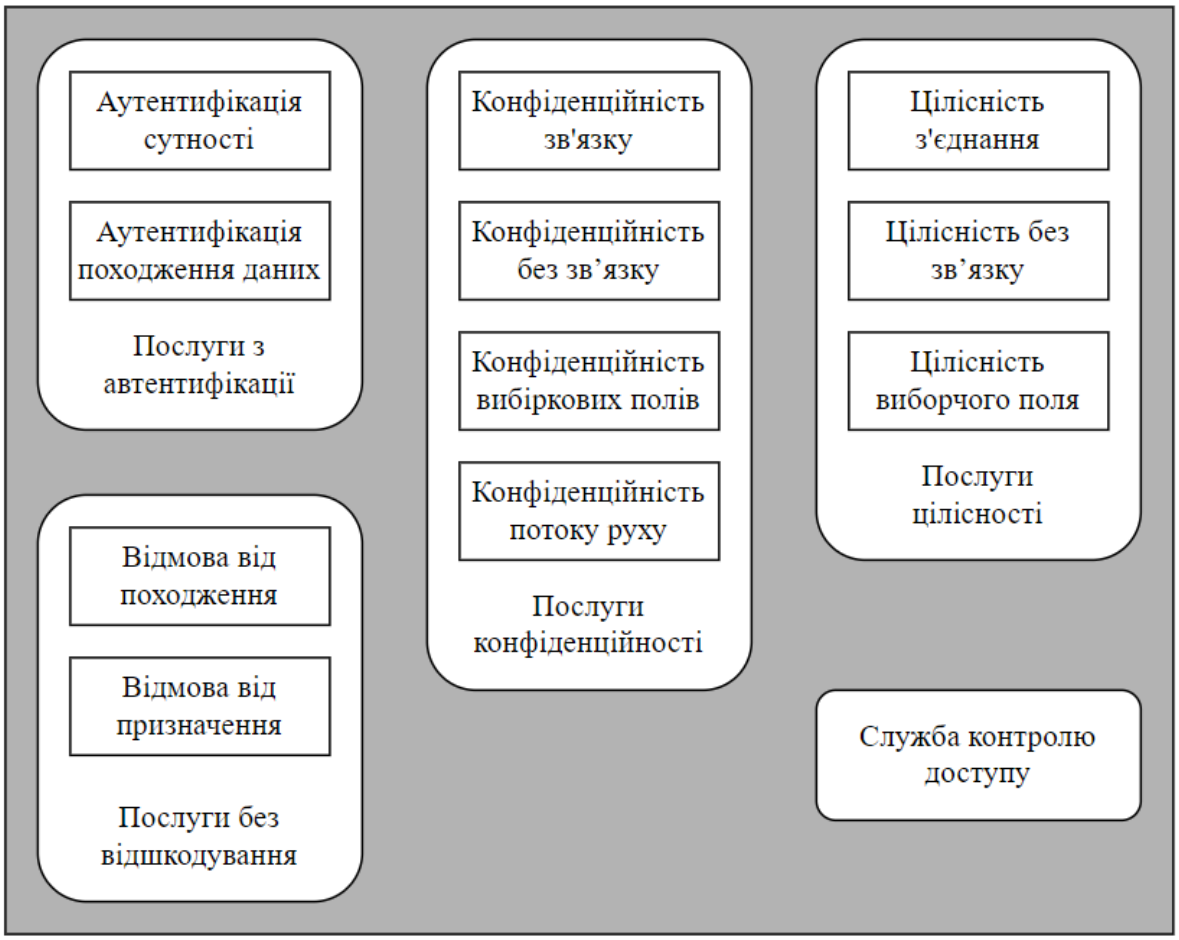

Рисунок 3.3 – Рівень функціональності служб безпеки IoT.

Служба доброчесності в цьому середовищі означає, що зміни повинні вноситись лише уповноваженими структурами та за допомогою

уповноважених механізмів. Послуги цілісності, включаючи цілісність з'єднання, цілісність без з'єднання та цілісність виборчого поля Послуги без відшкодування, включаючи відмову від походження та відмову від призначення та послугу контролю доступу є важливими для безпеки системи IoT. На рисунку 3.3 показано розділ рівня функціональності служб безпеки IoT.

3.2.3 Функція механізму захисту IoT

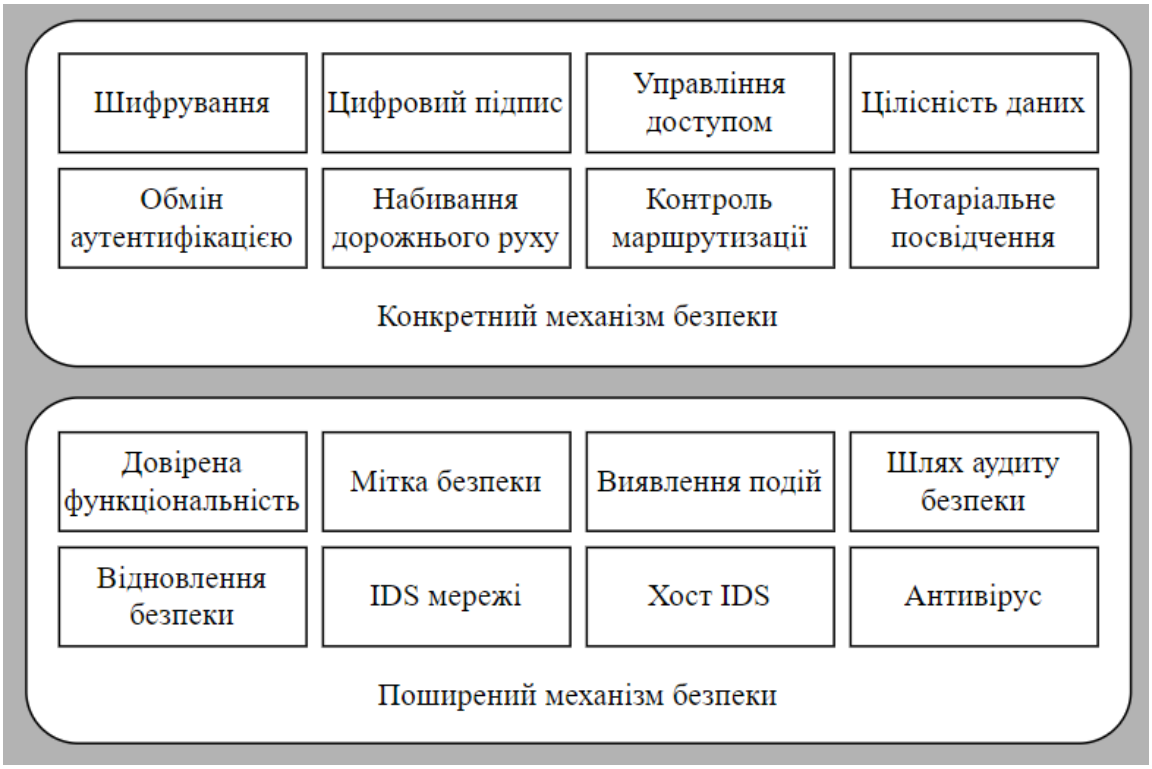

Рисунок 3.4 – Функціональний рівень механізмів захисту IoT.

Механізми безпеки забезпечують методи, алгоритми та схеми, необхідні для підтримки певних служб безпеки, визначених на рівні служб безпеки. Рівень функціональності механізму захисту IoT забезпечує механізми захисту як як специфічні механізми, так і загальні механізми. Конкретні механізми безпеки включають шифрування, цифровий підпис, контроль доступу, цілісність даних, обмін аутентифікацією, заповнення

трафіку, механізми контролю маршрутизації та механізми захисту нотаріуса. Поширені механізми безпеки включають надійну функціональність, мітку безпеки, рівномірне виявлення, слід аудиту безпеки, відновлення безпеки, мережеві і хостові IDS та антивірусні механізми безпеки [15].

На рисунку 3.4 показані модулі функціонального рівня механізму захисту IoT.

3.2.4 Основна функція безпеки IoT

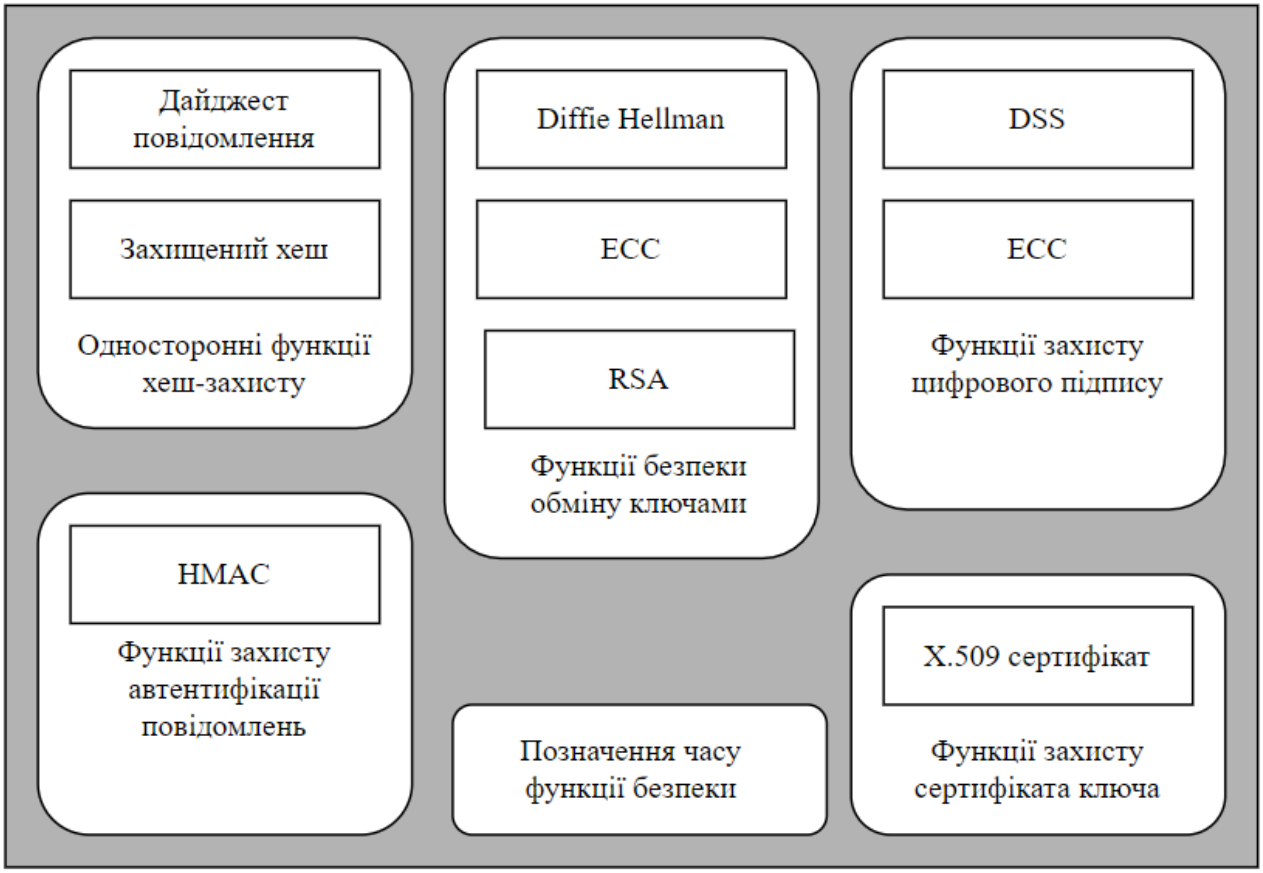

Рисунок 3.5 – Фундаментальний рівень функціональної безпеки IoT.

Основною властивістю IoT SMS є використання як комплексний автономний сервер безпеки, оскільки він може одночасно забезпечувати захист декількох додатків. Таким чином, вважається, що найнижчий рівень функціональності включає різні загальні арифметичні та модулі шифрування.

Функція фундаментальної безпеки IoT забезпечує основні функції безпеки, такі як односторонній хеш, дайджест повідомлень та захищені алгоритми хешування. Функції безпеки обміну ключами, включаючи Діффі Хеллмана, еліптичну криву та алгоритми RSA, включені в цей рівень. Цей рівень може включати стандарт цифрового підпису та алгоритми еліптичної кривої, автентифікацію повідомлень, код автентифікації, відмітку часу та сертифікати, включаючи стандарт сертифікату X.509. Цей рівень охоплює всі необхідні криптографічні елементарні функції для роботи IoT SMS. На рисунку 3.5 показані модулі рівня функціональної безпеки IoT безпеки.

3.3 Інформаційна база управління безпекою IoT

IoT SMIB є важливою складовою IoT SMS. Ця інформаційна база даних повинна бути структурована для підтримки впровадження всіх служб безпеки IoT в обчислювальному середовищі або середовищі спілкування. Інформаційна база управління безпекою IoT - це концептуальні сегменти ідентифікаторів розумних датчиків, профілів користувачів, списку контролю доступу та журналів безпеки. Зверніть увагу, що ця концепція не пропонує жодного змісту або форми для зберігання інформації. На рисунку 3.6 показано сегмент IoT SMIB.

Як показано, SMIB IoT – це сховище всієї інформації про вміст та параметрів, необхідних для нормального функціонування системи IoT. Існують взаємодії всередині і між рівнями SMS IoT та SMIB IoT.

Рисунок 3.6 – Сегменти SMIB IoT.

3.4 PKI для IoT Security

Робоча група з питань Інтернет-інженерії (IETF) створила інфраструктуру з відкритим ключем (PKI), щоб надати кілька та модель довіри. Важливими функціями PKI, що стосується безпеки системи IoT, є видача сертифікатів X.509, зберігання та оновлення ключів, надання послуг ряду протоколів та забезпечення контролю доступу.

PKI забезпечує базову основу для захисту інформації в комунікаціях за допомогою шифрування та автентифікації. За наявності PKI система IoT не вразлива до грубої сили та зловмисних атак. PKI забезпечує цілісність даних,

зібраних датчиками та інтелектуальними пристроями, а також забезпечує доступність, а також доступ до протоколу та конфігурації програми. Також PKI забезпечує конфіденційність рівня елементів у системі IoT.

У системі IoT дані шифруються вузлами бездротової мережі датчиків (WSN) і передають на шлюз. Шлюз розшифровує дані, а потім шифрує зведені дані перед передачею на верхні шари. Згідно із загальноприйнятими підходами, ключ шифрується серед усіх датчиків під час шифрування зібраних даних. Якщо ключ порушено, порушена вся система.

PKI надає пару відкрито-приватних ключів, математично пов'язаних. Якщо для шифрування даних використовується один ключ, розшифрувати дані може лише інший пов’язаний ключ. У випадку з елементарним рівнем у системі IoT дані, зібрані датчиками та інтелектуальними пристроями, шифруються за допомогою відкритого ключа, а потім за допомогою приватного ключа для дешифрування.

3.5 Переваги модульної системи управління безпекою для IoT

Система управління безпекою IoT забезпечує модульну структуру, яка забезпечує безліч служб безпеки та безліч механізмів безпеки. Таким чином, він реалізує вимоги безпеки для постачальників, постачальників мереж та виробників пристроїв. Різний модуль управління службами безпеки може викликати різні модулі механізмів безпеки, реалізуючи ефективну фундаментальну функцію безпеки для встановлення оптимальних вимог до безпеки та управління мережевою системою IoT.

Виходячи з вимог безпеки користувачів, модульна система управління безпекою IoTSMS реалізує ефективні методи та схеми безпеки в мережі IoT. Запропонований IoTSMS може забезпечити нові засоби безпеки, а також нові технології та технології. Він забезпечує загальну платформу для безпеки в системному середовищі IoT.

3.6 Сценарій управління безпекою IoT

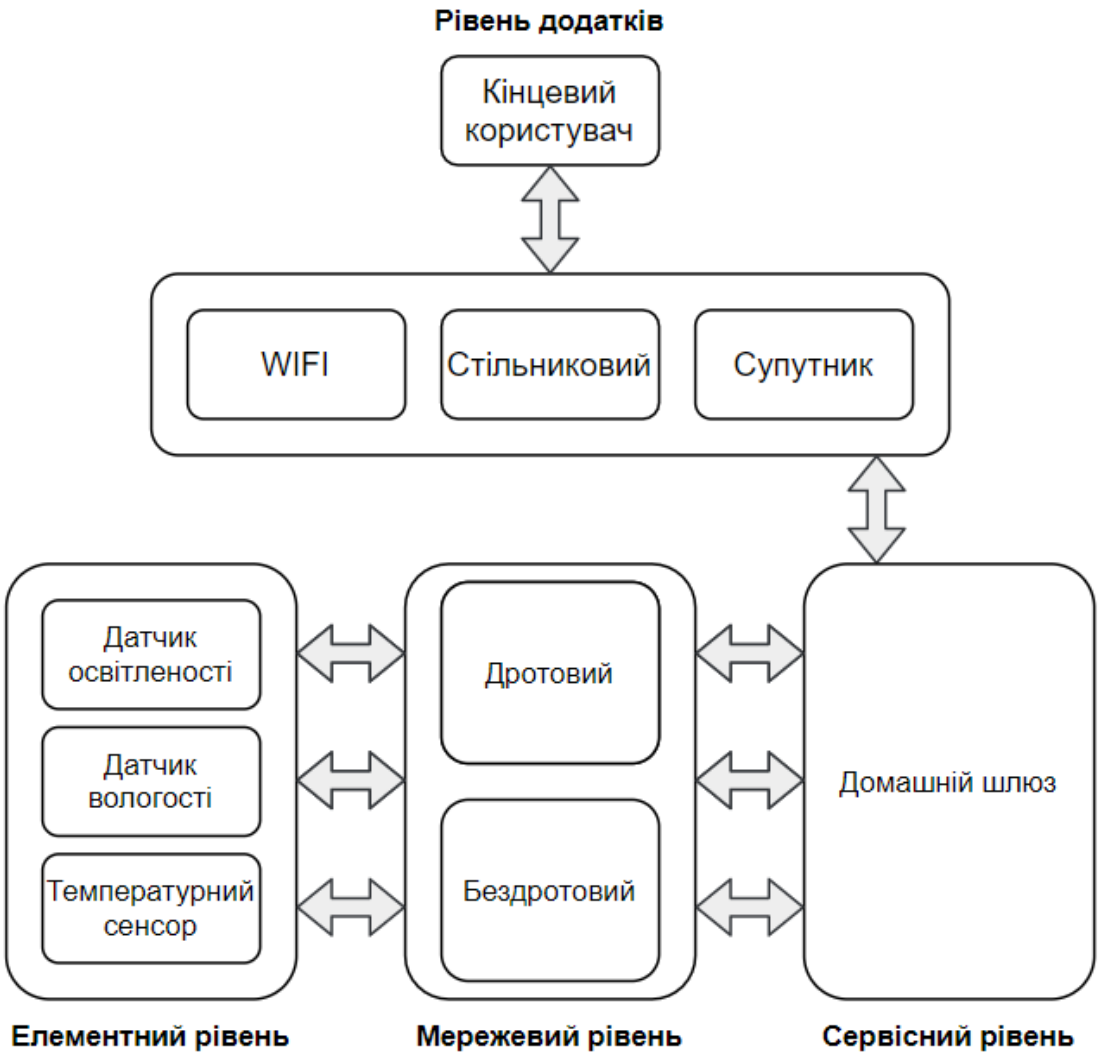

Рисунок 3.7 – Концепція сценарію розумного будинку

Давайте розглянемо розумний сценарій управління безпекою будинку. Власник будинку хоче стежити за рівнем комфорту у своєму будинку. Однак власник будинку знаходиться на своєму робочому місці і хоче за допомогою свого смартфона контролювати температуру, вологість, а також освітленість у своєму будинку.

На рисунку 3.7 показано можливий сценарій спілкування для цієї концепції розумного будинку.

3.7 Протоколи, використані у сценарії IoTSMS

Щоб проілюструвати складність служб безпеки в середовищі IoT, ми розглянемо практичний сценарій розумного будинку. Ми пояснюємо кожну задіяну функцію та необхідні дані на кожному рівні. Щоб задовольнити вимоги малопотужних та низькошвидкісних інтелектуальних пристроїв на елементарному рівні, ми повинні використовувати протокол бездротового зв'язку IEEE 802.15.4, щоб забезпечити необхідні послуги безпеки щодо конфіденційності, автентифікації та цілісності.

На мережевому рівні нам потрібно використовувати протокол 6LoPWAN, який реалізує механізм маршрутизації мереж із низьким енергоспоживанням та втратами. Механізм маршрутизації реалізує AES із 128-бітними ключами для MAC та підтримує RSA із SHA-256 для цифрових підписів для забезпечення служб безпеки конфіденційності та цілісності.

На рівні програми нам потрібно використовувати протокол CoAP, який працює над UDP, щоб зменшити вимоги до пропускної здатності та підтримати обмежені ресурсами пристрої та пристрої з низьким енергоспоживанням. Протокол CoAP забезпечує модель зв'язку "запит і відповідь" між кінцевими точками і приймає AES як криптографічний алгоритм для надання вищезазначених служб безпеки. На рисунку 3.8 показано еталонні протоколи, що використовуються в цьому сценарії розумного будинку, а в таблиці 3.1 – порівняння існуючих протоколів безпеки в сценарії розумного будинку.

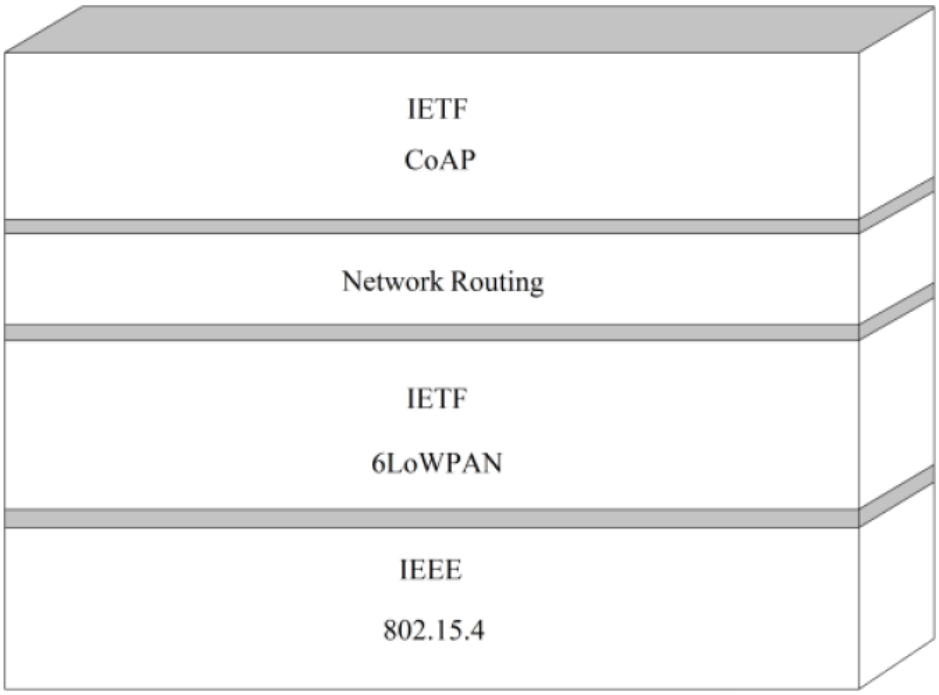

Рисунок 3.8 – Протоколи, що використовуються в розумному домі

Таблиця 3.1 – Порівняння протоколів безпеки в сценарії розумного будинку.

Шари		Протоколи		Служби безпеки	
Шар додатків	IETF CoAP	Конфіденційність	Аутентифікація	Чесність	Не відмова
Сервісний рівень	Маршрутизація IEEE	Конфіденційність	Аутентифікація	Чесність	Управління ключами
Мережевий рівень	IETF 6LoWPAN	Конфіденційність	Аутентифікація	Чесність	Доступність
Елементний шар	IEEE 802.15.4	Конфіденційність	Аутентифікація	Чесність	Управління доступом

З таблиці 3.1 видно, що більшість служб безпеки, необхідних на кожному рівні, однакові, за винятком різних механізмів, які будуть використовуватися на практиці.

3.8 Потік даних сценарію розумного будинку

Розглядаючи відповідний модуль служби безпеки, модуль механізму захисту та основний модуль базового захисту, ми можемо з’ясувати потік даних у сценарії управління безпекою розумного будинку:

- дані інформації навколишнього середовища, такі як температура, концентрація пари та інтенсивність світла, збираються різними датчиками. Потім дані обробляються за допомогою односторонньої хеш-функції для створення повідомлення цифрового підпису, яке викликається модулем управління службою автентифікації на рівні елементів. Протокол IEEE 802.15.4 реалізує механізм криптографії симетричного ключа AES для шифрування даних у режимі CCM за допомогою коду автентифікації повідомлень та коду цілісності повідомлень у 32-розрядних ключах;

- дані, що надходили з рівня елементів, були зашифровані за допомогою функції симетричного шифрування, яка викликається модулем управління цілісністю даних на мережевому рівні. Протокол 6LoWPAN реалізує механізм маршрутизації мереж з низьким енергоспоживанням та мереж з втратами (RPL), який реалізує AES із 128-бітними ключами, що забезпечує конфіденційність та послуги безпеки цілісності;

- службовий рівень отримав зашифровані дані, а модуль управління послугами доступності викликає функціональний модуль системи виявлення вторгнень мережі (NIDS) для запобігання атаці DoS під час передачі через Інтернет, WIFI або стільникову мережу;

- модуль управління службою автентифікації на рівні програми викликає модуль центру сертифікації ключів для перевірки ідентичності користувача шляхом порівняння профілю користувача. Потім користувач розшифровує повідомлення за допомогою закритого ключа, який надається модулем PKI. Протокол CoAP забезпечує модель зв'язку "запит і відповідь" між кінцевими точками і використовує AES як криптографічний алгоритм симетричного ключа для забезпечення конфіденційності служб безпеки;

- тепер користувач може використовувати інтерфейс прикладного програмування (API) на смартфоні для дистанційного контролю температури, вологості, а також освітленості в будинку під ефективним захистом безпеки.

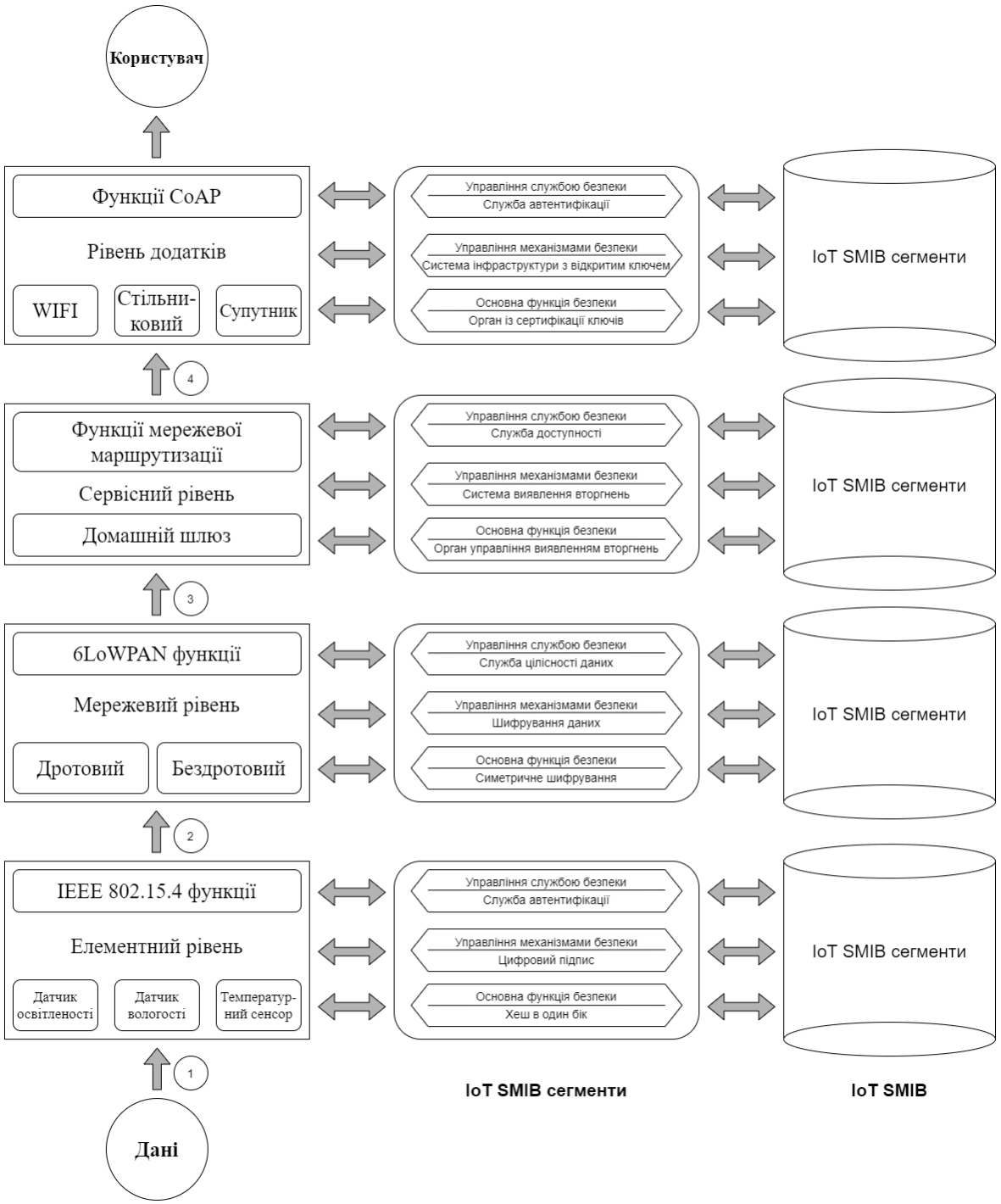

Рисунок 3.9 – Потік даних сценарію розумного будинку.

На рисунку 3.9 показано потік даних сценарію розумного будинку. На діаграмі показано управління, яке об'єкти IoT беруть участь у кожному рівні для розгляданого сценарію розумного будинку.

Наведений вище сценарій ілюструє спрощене застосування. У випадку зловмисників та труднощів може виникнути низка потенційних проблем, які потребують вирішення. Це легко вирішити, якщо запропонований IoTSMS є на місці.

4 МОДЕЛЬ ЗАХИСТУ ПРИСТРОЇВ В IOT

4.1 Впровадження запропонованої шаруватої моделі Cloud-Edge-IoT

Запропонований підхід полягає у забезпеченні заходів безпеки, встановлених перед розгортанням пристроїв з підтримкою IoT, у захищеній мережі та забезпеченні безпечного спілкування та обміну даними для захисту конфіденційності даних за допомогою шифрування. На рисунку 4.1 нижче показано абстракцію апаратного, програмного забезпечення та моделі зв'язку. Модель складається з хмари AWS як основної хмари, Raspberry Pi 4 як Edge Node та віртуальних машин як пристроїв IoT. Створена система з платним обліковим записом AWS, має повний доступ до ресурсів, що надаються AWS, включаючи сертифікати та ключі шифрування, авторизацію та автентифікацію.

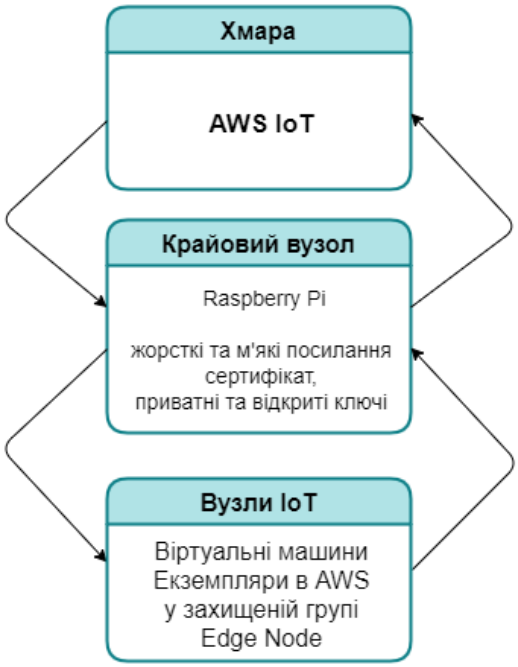

Рисунок 4.1 – Запропонована модель системи.

З наявних ресурсів AWS ми використали веб-службу AWS Identity and Access Management (IAM). Це дозволило контролювати доступ користувача шляхом ініціалізації облікового запису IAM (облікового запису користувача) для кожного користувача. З міркувань безпеки не було використано кореневий обліковий запис AWS, але було ініціалізовано користувача IAM з адміністративними дозволами. Щоб налаштувати Raspberry Pi як вузол Edge (AWS Greengrass Core), AWS Greengrass Core взаємодіє безпосередньо з хмарою і працює локально. Raspberry Pi було налаштовано шляхом додавання функцій захисту жорстких та м'яких посилань Linux. Щоб встановити зв'язок між AWS та Raspberry було використано AWS Greengrass Core, щоб створити групу з основним пристроєм, а також усі інші пристрої IoT, щоб дозволити їм спілкуватися з краєм.

Мною було створено сертифікати для автентифікації всіх пристроїв за допомогою AWS, приватні та відкриті ключі для безпечного з'єднання з фронтом та theAWS. Основні сертифікати були сформовані AWS після того, як я створила групу Greengrass, як показано на рисунку 4.2 нижче. Завантажила згенеровані файли в Raspberry Pi і запустили ядро Greengrass.

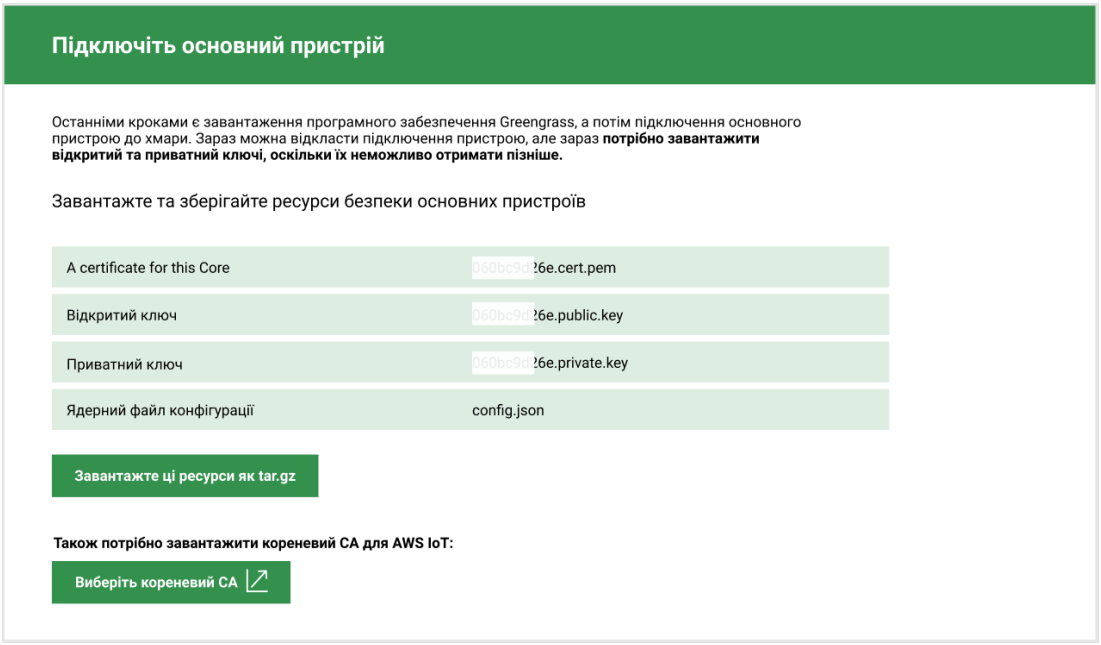

Рисунок 4.2 – Сертифікат, приватні та відкриті ключі

4.2 Обговорення та аналіз

Було створено простий сценарій, зробивши два пристрої з підтримкою IoT для взаємодії між собою через мій край. Пристрої IoT були налаштовані як віртуальні машини, створені в AWS, і додані до ядра Greengrass, як показано на 4.3 нижче.

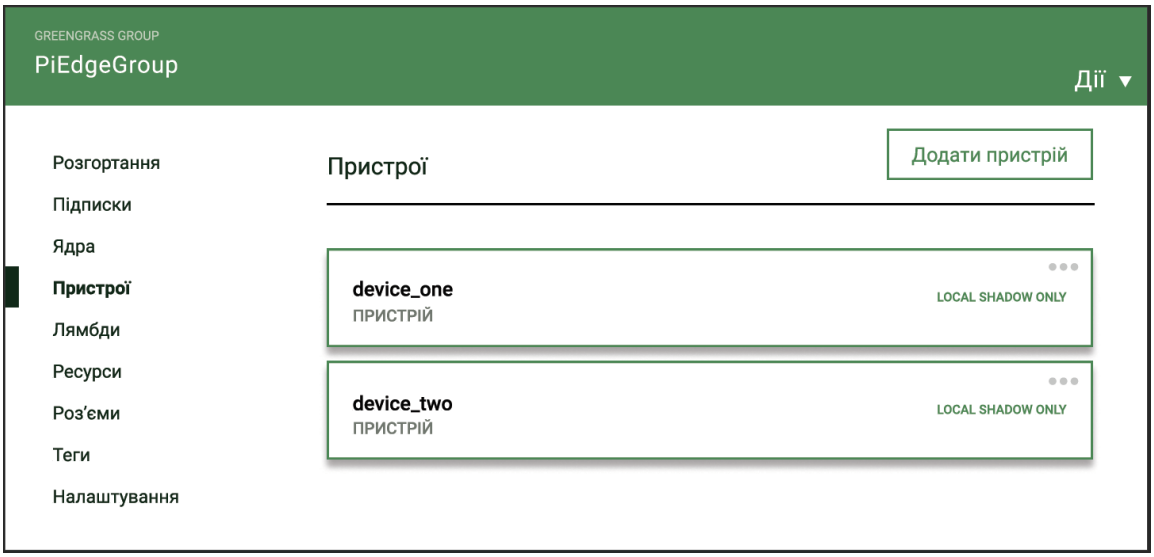

Рисунок 4.3 – Вузли з підтримкою IoT

Під час створення для кожного пристрою створюються спеціальні сертифікати, відкриті та закриті ключі для їх автентифікації за допомогою AWS та за допомогою пристрою Greengrass Core. Зв'язок між цими двома пристроями здійснювався через захищений механізм за допомогою протоколу MQTT, який називається посередником повідомлень. Нарешті, на рисунку 4.4 показано як вузли IoT, так і Edge-вузол успішно зв’язані, а обмін даними завершено в певній точці.

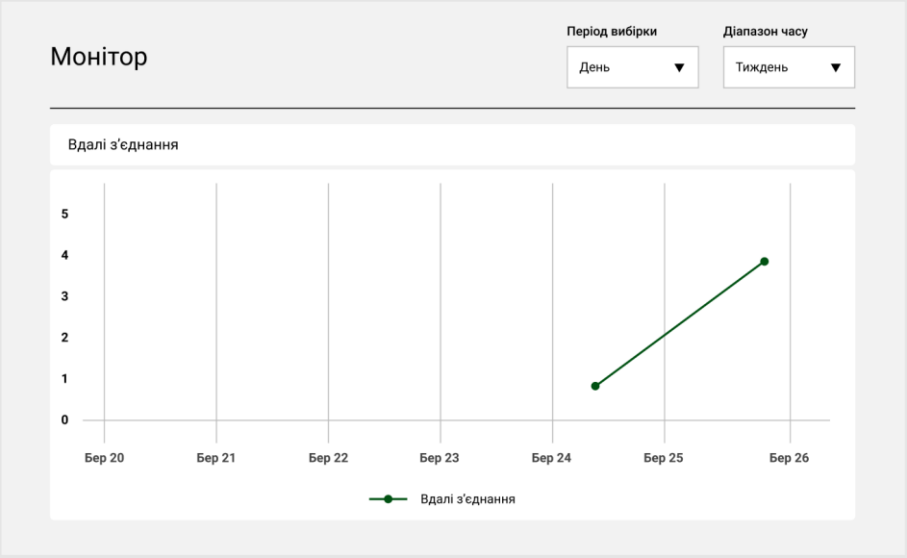

Рисунок 4.4 – Успішне з’єднання та обмін даними між вузлами. Вісь x представляє дні місяця, а вісь y - кількість з'єднань.

Нижче наведено основні моменти, які слід спостерігати щодо робочого середовища AWS та реалізованої моделі:

- у загальній моделі пристрої IoT з'єднуються між собою або з хмарою через AWS IoT Core;

- у запропонованій моделі було додано крайову концепцію, використавши основну концепцію IoT Greengrass в AWS, і представлено її за допомогою Pi, щоб була змога уявити її як додатковий посередник між пристроями IoT та Core AWS IoT, а потім хмарою;

- кожен пристрій потребує свого сертифіката, приватного ключа та кореневого сертифіката CA (це сертифікат AWS IoT). Існують різні типи кореневого сертифіката CA залежно від типів пристроїв IoT;

- кожному пристрою потрібна політика, ця політика визначає, які операції може виконувати цей пристрій (підключення / отримання / публікація / передплата тощо).

Тому було створено пристрій, політику та сертифікат. Потім прикріплено політику до сертифіката, а потім прикріплено сертифікат до пристрою. Політика за замовчуванням показана на рисунку 4.5 нижче:

Рисунок 4.5 – Політика пристроїв за замовчуванням у Amazon Web Service (AWS).

- політика за замовчуванням передбачає, що пристрій може виконувати всі дії (Action: iot: *) з усіх інших пристроїв та з ними (Resource: *);

- у нашій моделі було створено модифіковану політику для обробки доданого шару Greengrass;

- крім того, дія: greengrass: * означає, що пристрій у групі Greengrass може виконувати всі дії від та до інших пристроїв тієї самої групи Greengrass (Resource: *).

Модифікована політика для запропонованої моделі показана на рисунку 4.6. У сценарії зв'язок здійснюється за допомогою протоколу MQTT, який є протоколом від машини до машини. MQTT використовується, оскільки він легкий (повідомлення невеликого розміру і потребує низької потужності), тому він підходить для обмеженого середовища (датчики як приклад у реальних додатках).

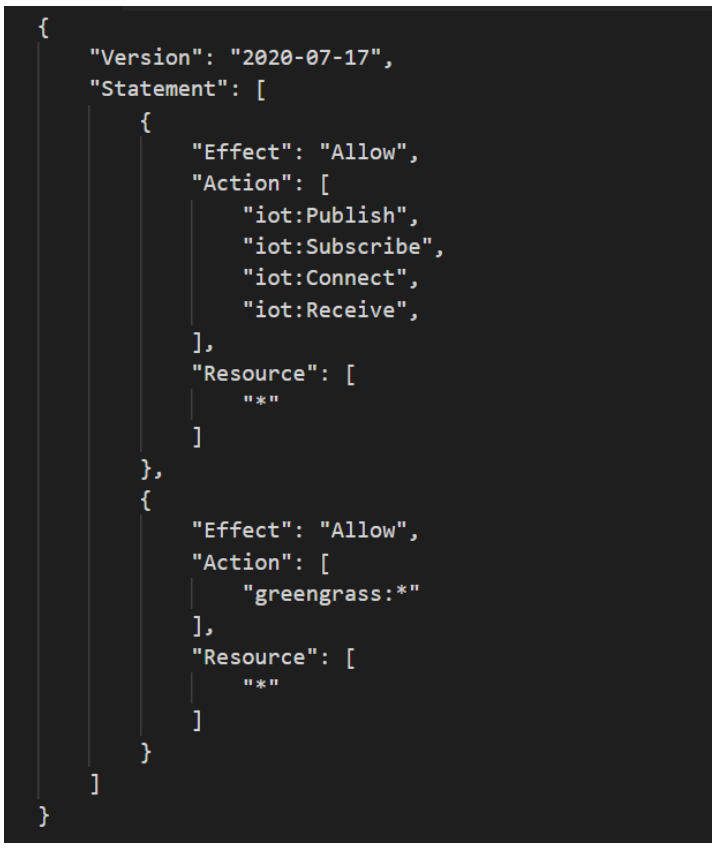

Рисунок 4.6 – Модифікована політика пристрою для включення крайового шару в запропоновану модель.

Більше того, важливо знати, що пристрої IoT в AWS емулюються як клієнти MQTT (якщо вони віртуальні - як це відбувається в моєму сценарії), а клієнти MQTT спілкуються через Тему MQTT.

З'єднання можна уявити як захищений канал між клієнтами, воно створюється, тоді клієнти можуть на нього підписатися, а інші клієнти публікують на ньому повідомлення.

Тепер процес налаштування Raspberry Pi включає встановлення JAVA JDK8, файлів Greengrass, на додаток до відповідного програмного забезпечення Core (залежно від використовуваного пристрою - у нашому випадку це Raspberry Pi 4).

Всі ці файли були передані в Raspberry Pi 4, як показано на рисунку 4.7 нижче.

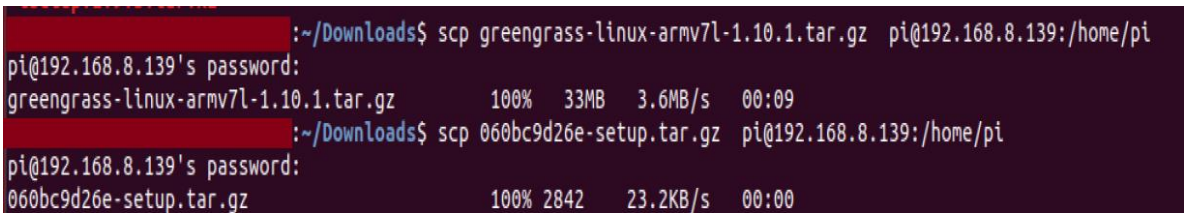

Рисунок 4.7 – Налаштування набору Raspberry Pi 4.

Після передачі необхідних файлів на Raspberry Pi 4 мені потрібно було їх витягти та внести деякі зміни в деякі файли конфігурації, щоб відповідати створеним сертифікатам та ключам.

Нарешті, було запущено основний пристрій Greengrass. Рисунок 4.8 нижче показує, що пристрій Raspberry успішно працював як Edge.

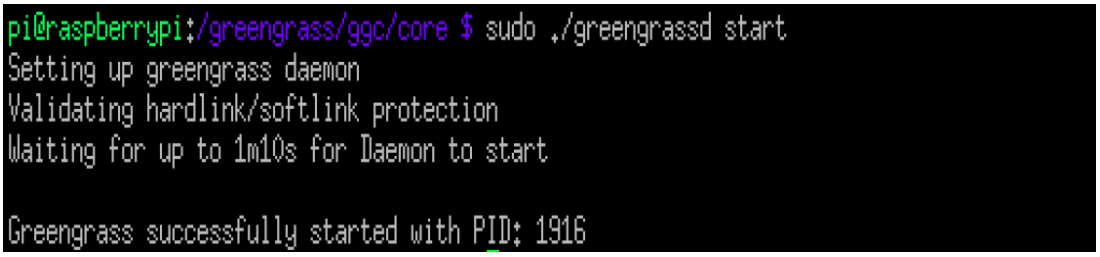

Рисунок 4.8 – Успішний запуск Greengrass на наборі Raspberry Pi 4.

Налаштувавши середовище та переконавшись, що воно готове, я створила тему MQTT у сценарії та назвала її (моя / тема). Потім створила пристрій для підписки на мою / тему та створила інший пристрій, який буде видавцем моєї / теми. Усі пристрої можуть виконувати всі дії з усіма іншими пристроями (політика за замовчуванням), і всі повідомлення обмінюються успішно. На рисунку 4.9 показано різні типи обміну повідомленнями за один день. На час з'єднання впливає багато факторів, включаючи затримку мережі та використовувану платформу.

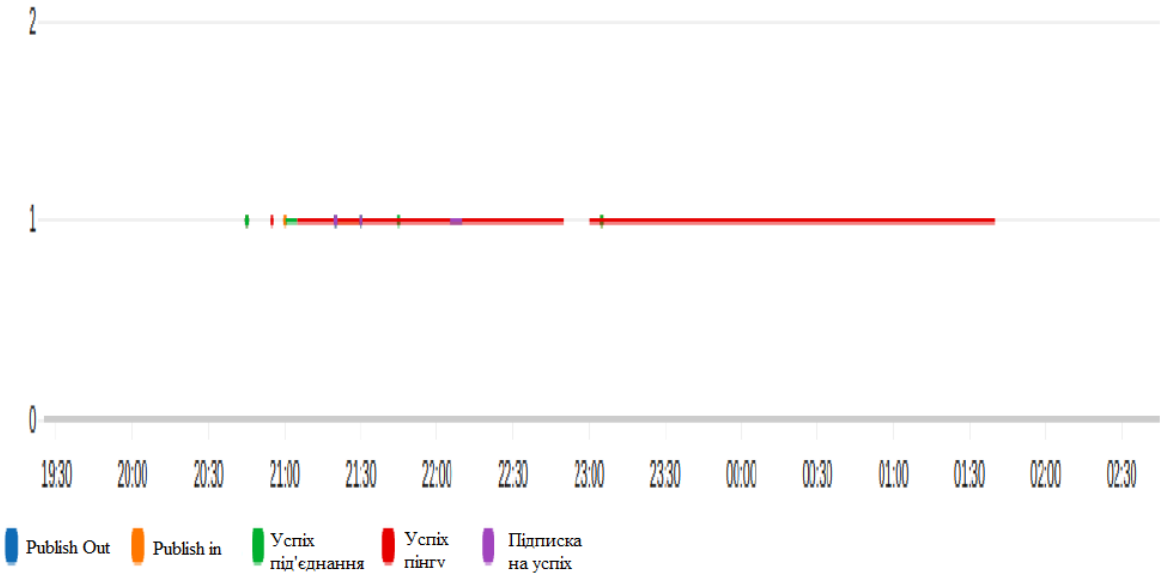

Рисунок 4.9 – Успішно обмінювалися повідомленнями різних типів: PublishOut Success, PublishIn Success, успіх під’єднання, успіх пінгу, підписка на успіх. Вісь x представляє години дня, а вісь y – один день.

Запропонована модель IoT показала, що можна забезпечити заходи щодо конфіденційності та безпеки, встановлені до того, як дозволено пристрою чи вузлу, що підтримує IoT, обмінюватися даними або ділитися ними. Після успішного впровадження та конфігурації я впевнена, що мої активи захищені. Описана модель у цій роботі може бути використана для забезпечення захищених середовищ і систем IoT із обчислювальними шарами туману / краю та злиттям датчиків. Багато реальних програм можуть використовувати цю модель, такі як охорона здоров’я, військова справа, аварійне відновлення та багато інших. Давайте розглянемо справу з охороною здоров’я; наприклад, використовуючи запропоновану модель, засновану на політиці, користувачі зможуть довіряти своєму медичному працівнику, щоб забезпечити їм захист, щоб вони знали, що за ними доглядають. Медичні компанії інвестують у носні засоби, вірячи, що вони допоможуть підвищити продуктивність робочої сили, скоротити прогули та зменшити витрати на охорону здоров'я. Іншим значущим фактором

пристроїв, що носяться, є переконання, що це може дати людям з обмеженими можливостями. Наприклад, людина з особливими потребами зможе вводити команди та текст, скажімо, просто рухаючи пальцем вгору та вниз. Остаточним методом, але не обмежуючись цим, є кількість користувачів безпеки, які можуть застосувати до своїх облікових записів. Наприклад, люди можуть обмежити, хто може переглядати їхні публікації в соціальних мережах або правила, що пояснюють важливість більшої безпеки для їхнього облікового запису (тобто двофакторна автентифікація).

ВИСНОВКИ

Пристрої та програми IoT відіграють важливу роль у нашому сучасному житті. Ми можемо бачити пристрої IoT майже скрізь з наших будинків, офісів, торгових центрів, шкіл, аеропортів та багатьох інших місць, щоб надати нам безпечні послуги на замовлення.

Пристрої IoT підтримують співпрацю із зацікавленими сторонами та допомагають зрозуміти бізнес-вимоги та результати. Крім того, аналітика та обробка даних на основі IoT можуть підвищити продуктивність та ефективність промислової інфраструктури

Більше того, системи IoT застосовують різні види корисних технологічних досягнень у різних секторах. Багато постачальників та компаній застосовують величезну кількість політик захисту своїх підключених пристроїв від шкідливих атак. Оскільки більша частина цих пристроїв підключається до наших приватних мереж та Інтернету, повідомляється про більше проблем щодо конфіденційності та безпеки. Ми читаємо і чуємо, що наша кавова машина шпигує за нашими розмовами; наш розумний дзвінок надсилає наші гостьові фотографії до державних установ. Багато реальних прикладів наголошують на серйозності вразливостей безпеки, пов'язаних із використанням пристроїв IoT.

У цій роботі було запропоновано нові багатошарові моделі IoT: загальні та розширені з ідентифікацією компонентів конфіденційності та безпеки та шарів. Запропонована система IoT, що підтримується хмарою, була впроваджена та оцінена. Нижній рівень представлений IoT-вузлами, створеними з AmazonWeb Service (AWS), як віртуальні машини. Середній шар (Edge) реалізований як апаратний комплект Raspberry Pi 4 за підтримки середовища Greengrass Edge в AWS. Верхній шар, який є хмарою, реалізований із використанням хмарного середовища IoT в AWS. Протоколи безпеки та критичні сеанси управління знаходились між кожним із цих

рівнів, щоб забезпечити конфіденційність інформації користувачів. Було впроваджено сертифікати безпеки, щоб дозволити передачу даних між шарами запропонованої моделі IoT із підтримкою Cloud / Edge.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Безпека IoT. Чому безпека IoT повинна бути комплексною. // https://www.anti-malware.ru/

2. Інтернет речей. Що це та чому це важливо. // https://www.sas.com/

3. Інформаційна безпека інтернету речей// https://www.tadviser.ru/

4. Овідіу Вермесан, Пітер Фріс, Інтернет речей: збіжні технології для розумного середовища та інтегрованої екосистеми. Ольборг, Данія: Річкові видавці, 2013.

5. Punit Gupta, Jasmeet Chhabra, “Інтелектуальний дизайн будинку на основі IoT за допомогою управління енергією та безпекою”, Міжнародна конференція з інновацій та викликів у кібербезпеці, с. 6-10, серпень 2016 р.

6. М.У. Фарук, Мухаммед Васім, Анжум Хайрі, Садія Мазхар, “Критичний аналіз проблем безпеки Інтернету речей (IoT)”, Міжнародний журнал комп’ютерних програм, вип. 111, с. 1-4, лютий 2015 р.

7. Овідіу Вермесан, Пітер Фріс, Інтернет речей від досліджень та інновацій до розгортання ринку. Ольборг, Данія: Річкові видавці, 2014.

8. Клаус Фінкенцеллер, Довідник RFID "Основи та програми в безконтактних смарт-картах", "Ідентифікація радіочастот і ближнє поле зв'язку". Уілтшир, Великобританія: John Wiley & Sons, 3-е видання, 2010.

9. Моуніб Ханафер, Мусін Геннун, Хуссей Т. Муфта, “Огляд протоколу MAC IEEE 802.15.4, що підтримує маяки, у бездротових сенсорних мережах”, IEEE Communication Survey & Tutorials, vol. 16, с. 856-876, грудень 2013 р.

10. Санія Вохра, Рохіт Шрівастава, "Опитування методів забезпечення 6LoWPAN", П'ята міжнародна конференція з систем зв'язку та мережевих технологій, с. 643-646, квітень 2015 р.

11. Василіос Карагіанніс, Періклі Чацімісіос, Франциско Васкес-Гальєго, Ісус Алонсо-Зарате, “Опитування протоколів рівня додатків для

Інтернету речей", Трансакція з IoT та хмарних обчислень, с. 1-8, квітень 2015 р.

12. Давіде Конзон, Томас Болоньєсі, Паоло Бріцці, Антоніо Лотіто, Ріккардо Томасі, Мауріціо А. Спіріто, "Архітектура, що базується на XMPP, для безпечних комунікацій IoT", Інтерактивна конференція з комп'ютерних комунікацій та мереж, с. 1-6, серпень 2012 р. .

13. Айкатеріні Мітрокоца, Мелані Р. Рібек та Ендрю С. Таненбаум, "Класифікація атак RFID", Журнал досліджень та інновацій, вип. 12, с. 491-505, листопад 2010 р.

14. Доктор Г. Падмаваті, пані Д. Шанмугапрія, "Огляд атак, механізмів безпеки та викликів у бездротових сенсорних мережах", Міжнародний журнал комп'ютерних наук та інформаційної безпеки, том 4, с. 2-7, вересень 2009 рік.

15. Рабі Прасад Падхі, Манас Ранджан Патра, Суреш Чандра Сатапаті, "Хмарні обчислення: проблеми безпеки та виклики дослідження", Міжнародний журнал комп'ютерних наук та інформаційних технологій та безпеки, вип. 1, с. 13-18, грудень 2010 р.

16. Дяченко В.О., Польська Б.Ю. Методи захисту інформації в IoT // Проблеми інформатизації. Тези доповідей восьмої міжнародної науково-технічної конференції / Черкаси-Харків-Баку-Бельсько-Бяла, 2020 – Т1.– С.46