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

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2024 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“ Програмне забезпечення системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку”

Виконав здобувач вищої освіти
IV курсу, групи КБ-20
ОПП «Кібербезпека»
спеціальності 125 «Кібербезпека»
_____ Карпіков О.С.
«____» _____ 2024 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Улічев О.С.
«____» _____ 2024 р.
Рецензент _____
_____

м. Кропивницький

Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*

Кафедра *Кібербезпеки та програмного забезпечення*

Освітній ступінь *бакалавр*

Галузь знань ._12 _____*“*Інформаційні технології*”*_____

Спеціальність *125 “*Кібербезпека*”*

Освітньо-професійна (освітньо-наукова) програма _*“*Кібербезпека*”*_

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 17 » січня 2024 року

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

Карпікову Олегу Сергійовичу

(прізвище, ім’я, по батькові)

1. Тема роботи *Програмне забезпечення системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку*

2. Керівник роботи *Улічев Олександр Сергійович, канд. техн. наук, ст. викладач кафедри*

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 135-02 від 01.04.2024 року

3. Строк подання студентом роботи до захисту *19.05.2024 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою роботи є розробка програмного забезпечення системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку*

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

1. Призначення та область використання.

2.Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи кібербезпеки в промислову експлуатацію.

6. Висновки

6. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

Структурна схема системи кібербезпеки	*1 аркуш*
Функціональна схема системи кібербезпеки	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*

7. Дата видачі завдання « 17 » січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	18.01.2024 р.	
2.	Постановка задачі, оформлення ТЗ	22.01.2024 р.	
3.	Розробка моделі компонента	02.02.2024 р.	
4.	Розробка структур даних	13.02.2024 р.	
5.	Розробка алгоритмів зв’язку та відображення	18.02.2024 р.	
6.	Програмування алгоритмів	23.02.2024 р.	
7.	Оформлення ПЗ	11.03.2024 р.	
8.	Попередній захист роботи	19.05.2024 р.	

Дата видачі завдання
« 17 » січня 2024 р.

Підпис керівника
_____ Улічев О.С.
(прізвище та ініціали)

Завдання прийнято до виконання
« 18 » січня 2024 р.

Підпис здобувача
_____ Карпіков О.С.
(прізвище та ініціали)

АНОТАЦІЯ

Карпіков О.С. Програмне забезпечення системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку. 125 Кібербезпека. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення для системи кібербезпеки, призначене для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку.

Метою розробки є створення програмного забезпечення системи кібербезпеки, що забезпечує захист віддалених об'єктів та сповіщення про можливі загрози через бездротові канали зв'язку.

Результат роботи це програмна реалізація системи кібербезпеки, яка моніторить стан віддаленого об'єкту, виявляє аномалії та сповіщає користувача про потенційні загрози, з можливістю застосування захистних протоколів.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів.

Розроблено зручний інтерфейс користувача.

Програма може використовуватися на комп'ютерах з ОС Windows 10/11.

Програму розроблено в середовищі Visual C++ з використанням MFC.

Ключові слова: кібербезпека, захист віддалених об'єктів, бездротові сповіщення

ABSTRACT

Karpikov O.S. Cybersecurity system software for protecting a remote facility with wireless communication channel alerts. 125 Cybersecurity. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this bachelor's graduation qualification work, software for a cybersecurity system designed to protect a remote object with wireless communication notification has been developed.

The aim of the development is to create cybersecurity software that ensures the protection of remote objects and notifies about potential threats through wireless communication channels.

The result of the work is the software implementation of a cybersecurity system that monitors the status of a remote object, detects anomalies, and notifies the user about potential threats, with the possibility of applying protective protocols.

During the work on the software model, an analysis of existing hardware and software tools was carried out.

A user-friendly interface has been developed.

The program can be used on computers with Windows 10/11 operating systems.

The software was developed in the Visual C++ environment using MFC.

Keywords: cybersecurity, protection of remote objects, wireless notifications.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

GPS – глобальна навігаційна система

IoT – Internet of Things, інтернет речей

UNB – ультра-вузькосмугова модуляція

VPN – віртуальна приватна мережа

MFC – Microsoft Foundation Class library

SDI – інтерфейс одного документа

MDI – інтерфейс багатьох документів

ООП – об'єктно-орієнтоване програмування

TCP/IP – протокол управління передачею/інтернет-протокол

AppKey – ключ додатку

MIC – код цілісності повідомлення

AES-128 – Advanced Encryption Standard з 128-біт ключом

ВСТУП

Актуальність теми. З розвитком технологій і зростанням кількості віддалених об'єктів, які потребують захисту, питання кібербезпеки набуває все більшої актуальності. Особливо це стосується об'єктів, розташованих за межами міста, де традиційні методи зв'язку можуть бути недоступними або ненадійними. В умовах постійного зростання кількості кіберзагроз, забезпечення безпеки віддалених об'єктів стає критично важливим завданням.

Безпека віддалених об'єктів, таких як сонячні батареї, потребує застосування сучасних технологій, які забезпечують надійний та оперативний обмін даними. Бездротові технології зв'язку відіграють ключову роль у цьому процесі, оскільки вони дозволяють забезпечити безперебійний зв'язок і моніторинг стану об'єктів у реальному часі. Використання бездротових каналів зв'язку для оповіщення про небезпеку чи несправності забезпечує оперативне реагування та мінімізує ризики.

У контексті кібербезпеки, бездротові технології є особливо важливими, оскільки вони дозволяють не лише моніторити стан об'єктів, але й оперативно реагувати на потенційні загрози. Вибір та впровадження ефективних бездротових технологій зв'язку забезпечує високий рівень безпеки, знижує ризики несанкціонованого доступу та підвищує надійність захисту віддалених об'єктів.

Таким чином, дослідження та розробка програмного забезпечення для системи кібербезпеки з оповіщенням бездротовим каналом зв'язку є надзвичайно актуальними. Це дозволяє створити надійні та ефективні рішення для захисту віддалених об'єктів, забезпечуючи їх безпеку та стабільну роботу в умовах сучасних викликів.

Мета й завдання дослідження. Метою роботи є розробка програмного забезпечення системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих бездротових технологій зв'язку для оповіщення та моніторингу віддалених об'єктів.

– Дослідження системи кібербезпеки з використанням бездротових технологій для захисту віддалених об'єктів.

– Програмна реалізація системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку.

Предмет та об’єкт дослідження. Об'єктом дослідження є віддалені об'єкти, що потребують кіберзахисту та моніторингу за допомогою бездротових технологій. До таких об'єктів належать, зокрема, сонячні батареї, вітрові турбіни, нафтові та газові установки, розташовані у віддалених місцевостях.

Предметом дослідження є програмне забезпечення для системи кібербезпеки, яке забезпечує захист віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку. Основна увага приділяється розробці методів та алгоритмів для забезпечення надійного зв'язку, моніторингу та оперативного реагування на загрози кібербезпеки.

Практична цінність отриманих результатів полягає в тому, що розроблене програмне забезпечення дозволяє успішно вирішувати задачі захисту віддалених об'єктів з оповіщенням бездротовим каналом зв'язку.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система захисту віддаленого об'єкта загалом призначена для забезпечення безпеки та контролю віддалених місць, таких як нафтові та газові установки, за допомогою сучасних технологій. Основні функції системи включають моніторинг стану обладнання, виявлення аномалій, та своєчасне сповіщення користувачів про потенційні загрози. Завдяки використанню бездротових сенсорів, системи аналізу даних в реальному часі та захищених каналів зв'язку, система надає можливість оперативно реагувати на критичні ситуації, мінімізуючи ризики та забезпечуючи стабільність роботи віддалених об'єктів.

У сучасному світі бездротові технології відіграють важливу роль у забезпеченні надійного зв'язку між різними об'єктами. Ці технології стають все більш поширеними завдяки їхній здатності забезпечувати швидкий та зручний обмін даними на великих відстанях без необхідності прокладання кабелів. Особливо це актуально для віддалених об'єктів, таких як сонячні батареї, вітрові турбіни, водяні насоси, які розташовані за межами міста і потребують постійного моніторингу та захисту.

Однією з ключових переваг бездротових технологій є їхня гнучкість і мобільність. Завдяки можливості легкої інтеграції з різними системами і пристроями, вони дозволяють оперативно реагувати на зміни в середовищі, забезпечуючи високу ступінь захищеності віддалених об'єктів. Це особливо важливо у контексті кібербезпеки, де швидке виявлення і реагування на загрози можуть запобігти значним втратам.

Згідно з прогнозами аналітичних агентств, використання бездротових технологій для захисту віддалених об'єктів буде лише зростати. Наприклад, за даними компанії MarketsandMarkets, світовий ринок бездротових систем безпеки

зросте до декількох мільярдів доларів до 2025 року, завдяки чому збільшиться кількість впроваджень таких систем у різних галузях.

Особливо перспективними є технології, які використовують сучасні протоколи зв'язку, такі як Wi-Fi, Zigbee, LoRaWAN та інші. Вони дозволяють забезпечувати стабільний та надійний зв'язок на великих відстанях з мінімальними енергетичними витратами. Такі технології дозволяють створювати системи, що здатні працювати тривалий час без необхідності частого обслуговування, що є важливим фактором для віддалених об'єктів.

Таким чином, розвиток і впровадження бездротових технологій у сфері кібербезпеки для захисту віддалених об'єктів є надзвичайно важливим і актуальним напрямком дослідження. Це дозволяє створити більш ефективні та надійні системи безпеки, які забезпечують безперебійний моніторинг і захист віддалених об'єктів у сучасних умовах постійно зростаючих кіберзагроз.

1.2 Область застосування

Бездротові технології мають широкий спектр застосувань у різних галузях завдяки їхній гнучкості, надійності та ефективності. Вони дозволяють забезпечити безперебійну передачу даних, зменшити витрати на прокладання кабелів та підвищити ефективність управління віддаленими об'єктами, що робить їх незамінними у сучасному світі. Усі сучасні країни йдуть в ногу з майбутнім та використовують ці технології заради полегшання процесу пильнувати та захищати віддалені об’єкти при доцільній потребі.

Бездротові технології та українські компанії

В Україні є кілька відомих компаній, які активно використовують бездротові технології для захисту віддалених об'єктів, а саме:

Ajax Systems - компанія, що заснована у 2011 році, спеціалізується на розробці та виробництві охоронних систем, які використовують бездротові технології для моніторингу та захисту віддалених об'єктів. Ajax Systems є міжнародною компанією з головним офісом у Києві, яка проектує та виготовляє

інноваційні системи безпеки, що забезпечують надійний зв'язок і контроль у реальному часі.

Wezom – компанія, яка пропонує комплексні послуги з кібербезпеки, включаючи проактивний захист від кіберзагроз, що стає все більш актуальним для бізнесів у різних сферах. Wezom надає індивідуальні рішення для забезпечення безпеки даних та запобігання несанкціонованому доступу до віддалених об'єктів за допомогою сучасних бездротових технологій.

DTEK - це одна з найбільших енергетичних компаній України, яка активно використовує бездротові технології для моніторингу та управління своїми об'єктами. DTEK забезпечує стабільний зв'язок і контроль за віддаленими енергетичними установками, що сприяє ефективному управлінню та безпеці цих об'єктів.

Kernel - велика агропромислова компанія, яка застосовує бездротові технології для моніторингу та управління своїми сільськогосподарськими угіддями. Використання таких технологій дозволяє компанії оптимізувати процеси зрошення та обробки ґрунту, забезпечуючи ефективність і безпеку своїх віддалених об'єктів.

Ці перелічені компанії гарно демонструють, як бездротові технології можуть ефективно використовуватися для забезпечення безпеки та управління віддаленими об'єктами в різних галузях, від енергетики до сільського господарства та нерухомості.

Бездротові технології в сільському господарстві

В наш час бездротові технології знаходять широке застосування в сільському господарстві, значно підвищуючи його ефективність та продуктивність. Наприклад, бездротові сенсори вологості ґрунту, розташовані на полях, дозволяють фермерам отримувати дані в реальному часі, оптимізуючи використання води для іригації. Це запобігає надмірному або недостатньому поливу, що є критично важливим для забезпечення здорового росту рослин. Дрони та GPS-технології допомагають створювати точні карти полів і моніторити стан посівів, виявляючи проблемні зони для вчасного реагування.

Крім того, автоматизовані системи іригації, оснащені бездротовими сенсорами та контролерами, можуть автоматично регулювати полив на основі даних про вологість ґрунту та погодні умови, що зменшує витрати на воду та підвищує ефективність використання ресурсів. У сфері тваринництва бездротові трекери допомагають відстежувати місцезнаходження та стан здоров'я худоби, що дозволяє фермерам своєчасно виявляти хвороби та інші проблеми. Бездротові метеостанції вимірюють параметри навколишнього середовища, такі як температура, вологість та опади, що допомагає в плануванні агротехнічних заходів.

Інтеграція бездротових сенсорів, трекерів і контролерів з IoT платформами дозволяє фермерам отримувати комплексні дані про всі аспекти їхнього господарства в реальному часі, сприяючи прийняттю більш обґрунтованих рішень. Наприклад, в Україні фермери в Херсонській області застосовують бездротові сенсори для моніторингу вологості ґрунту та автоматизації систем іригації, що дозволяє значно підвищити врожайність та ефективність використання води. Завдяки таким технологіям, як LoRaWAN, NB-IoT та Wi-Fi HaLow, сільське господарство стає більш стійким, ефективним та екологічно чистим.

В Україні бездротові технології активно використовуються в аграрному секторі. Наприклад, у Херсонській області фермери застосовують бездротові сенсори для моніторингу вологості ґрунту та автоматизації систем іригації, що дозволяє значно підвищити врожайність та ефективність використання води.

Бездротові технології: сонячні та вітрові електростанції

У галузі відновлюваної енергетики бездротові технології відіграють важливу роль, забезпечуючи ефективний моніторинг, управління та інтеграцію розподілених енергетичних ресурсів. Сонячні електростанції часто охоплюють великі території із розміщеними на значних відстанях фотоелектричними панелями. Для контролю їхньої продуктивності та технічного стану використовуються бездротові сенсорні мережі. Окремі сенсори, вбудовані в панелі, відстежують показники потужності, температури, освітленості тощо та

безпровідним шляхом передають зібрані дані на центральну систему управління.

Це дозволяє оперативно виявляти будь-які відхилення, несправності чи зниження ефективності окремих вузлів і вживати своєчасних заходів, максимізуючи загальну продуктивність сонячної електростанції. Крім того, бездротові технології застосовуються для керування системами стеження за сонцем - автоматичного позиціонування панелей для забезпечення оптимального кута нахилу щодо сонячного випромінювання.

На вітрових електростанціях бездротові сенсори розміщуються безпосередньо на лопатях вітрогенераторів та решті конструкцій для моніторингу їхнього стану під час роботи. Зібрані дані про вібрації, навантаження, температуру тощо дозволяють своєчасно виявляти потенціальні ризики та планувати обслуговування вузлів, запобігаючи виходу устаткування з ладу.

Безпека передачі інформації є критично важливим аспектом у роботі цих електростанцій, оскільки ці об'єкти розташовані у віддалених місцях і можуть бути вразливими до кібератак. Використання сучасних протоколів безпеки, таких як шифрування даних та автентифікація пристроїв, допомагає захистити передані дані від несанкціонованого доступу та маніпуляцій. Наприклад, технологія LoRaWAN використовує шифрування AES-128 для захисту даних на всіх етапах передачі, що запобігає перехопленню та модифікації даних зловмисниками. Крім того, застосування механізмів автентифікації дозволяє переконатися, що доступ до мережі мають лише авторизовані пристрої, що підвищує загальну безпеку системи.

Бездротові технології в нафтових і газових установках

У нафтогазовій промисловості бездротові технології відіграють важливу роль, забезпечуючи ефективний моніторинг та управління розподіленими об'єктами. Одним із ключових застосувань є використання бездротових сенсорних мереж для контролю стану свердловин та обладнання. Сенсори, розміщені на бурових установках, трубопроводах та резервуарах, безпровідним шляхом передають дані про тиск, температуру, вібрацію та інші параметри на

центральні системи моніторингу. Це дозволяє оперативно виявляти будь-які відхилення та потенційні загрози, запобігаючи аваріям та несанкціонованим викидам.

Крім того, бездротові технології широко застосовуються для забезпечення безпеки персоналу на віддалених промислових майданчиках. Наприклад, працівники можуть бути оснащені персональними сенсорами, які відстежують їхнє місцезнаходження, параметри довкілля та стан здоров'я. У разі виникнення надзвичайної ситуації або порушення безпечних умов, система автоматично надсилає сигнал тривоги та інформацію для оперативного реагування.

Бездротові мережі також використовуються для віддаленого управління технологічними процесами на родовищах. Оператори можуть дистанційно контролювати та регулювати роботу свердловинного обладнання, клапанів, насосів тощо, отримуючи актуальні дані в режимі реального часу. Це підвищує ефективність виробництва та знижує ризики, пов'язані з безпосередньою присутністю персоналу на небезпечних ділянках.

Розвиток технологій Інтернету речей (IoT) сприяє впровадженню ще більш розвинених бездротових рішень у нафтогазовому секторі. Наприклад, можна створювати високошвидкісні бездротові мережі для передачі відеоданих з віддалених майданчиків, забезпечуючи постійний візуальний моніторинг та контроль процесів видобутку та транспортування вуглеводнів.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

LoRaWAN (Long Range Wide Area Network)

Це технологія бездротового зв'язку, яка використовується для передачі даних на великі відстані з мінімальним енергоспоживанням. Ця технологія розроблена спеціально для застосувань у сфері Інтернету речей (IoT), де необхідно забезпечити надійну передачу даних від великої кількості сенсорів і пристроїв, розташованих на значних відстанях один від одного. LoRaWAN використовує модуляцію LoRa, що дозволяє передавати дані на відстань до 15 км у сільській місцевості та до 5 км у міських умовах, забезпечуючи стабільний зв'язок навіть у важкодоступних місцях.

LoRaWAN підтримує двосторонню передачу даних, що дозволяє не лише збирати інформацію з сенсорів, але й відправляти команди управління до пристроїв. Це робить можливим автоматизоване управління системами, такими як іригаційні системи в сільському господарстві або системи моніторингу сонячних і вітрових електростанцій.

Безпека передачі даних у LoRaWAN забезпечується за допомогою шифрування AES-128. Кожен пристрій має унікальні ключі шифрування, що запобігає перехопленню і несанкціонованому доступу до даних. Крім того, мережеві сервери LoRaWAN використовують механізми автентифікації для перевірки пристроїв, що підключаються до мережі, забезпечуючи додатковий рівень захисту. Це робить LoRaWAN надійним вибором для застосувань, де необхідно забезпечити високий рівень безпеки передачі даних.

Wi-Fi HaLow (Wi-Fi 802.11ah)

Розширення стандарту Wi-Fi яке призначене для забезпечення довготривалого зв'язку з низьким енергоспоживанням, особливо підходить для додатків Інтернету речей (IoT). Wi-Fi HaLow працює в субгігагерцовому діапазоні частот, що дозволяє забезпечувати значно більшу дальність зв'язку порівняно з традиційними Wi-Fi мережами, які використовують діапазони 2,4 ГГц та 5 ГГц. Завдяки цьому Wi-Fi HaLow може досягати радіусу дії до 1 км, що робить його ідеальним для використання в промислових зонах, розумних містах і сільському господарстві.

Можливість підтримувати високу пропускну здатність, зберігаючи при цьому низьке енергоспоживання, є основною перевагою Wi-Fi HaLow. Пристрої, що використовують цю технологію, можуть працювати від батарей протягом тривалого часу, що є критично важливим для IoT-додатків, де потрібно забезпечити безперервну роботу без частих замін батарей. Завдяки своїм характеристикам, Wi-Fi HaLow може передавати дані зі швидкістю від 150 Кбіт/с до 15 Мбіт/с, що дозволяє використовувати його не лише для сенсорних додатків, але й для передачі відео високої роздільної здатності та інших даних з високою інтенсивністю.

Wi-Fi HaLow також забезпечує високу проникність сигналу через перешкоди, такі як стіни та інші об'єкти, завдяки використанню низьких частот. Це робить його ідеальним для використання в промислових зонах і будівлях з товстими стінами, де традиційний Wi-Fi може не забезпечувати достатнього покриття. Крім того, Wi-Fi HaLow підтримує високу щільність пристроїв, дозволяючи підключати тисячі пристроїв до однієї точки доступу, що є важливим для масштабних IoT-мереж.

Безпека передачі даних у Wi-Fi HaLow забезпечується за допомогою сучасних протоколів шифрування та автентифікації, таких як WPA3. Це гарантує захист даних від несанкціонованого доступу та атак, забезпечуючи надійний і безпечний зв'язок. Завдяки цьому Wi-Fi HaLow є відмінним вибором для додатків, де необхідно забезпечити високий рівень безпеки передачі даних,

наприклад, у розумних будинках, медичних закладах та промислових об'єктах.

NB-IoT (Narrowband IoT)

NB-IoT (Narrowband IoT) — це технологія бездротового зв'язку, спеціально розроблена для додатків Інтернету речей (IoT), які потребують передачі даних на великі відстані з низьким енергоспоживанням. NB-IoT працює в ліцензованих спектрах, що забезпечує високу надійність і захищеність зв'язку. Основною особливістю NB-IoT є його здатність підтримувати велику кількість пристроїв на невеликій площі, що робить цю технологію ідеальною для міських умов і промислових застосувань.

NB-IoT забезпечує довготривалу роботу пристроїв від батареї, завдяки низькому енергоспоживанню. Пристрої, що використовують цю технологію, можуть працювати від одного заряду батареї до 10 років. Це важливо для сенсорних мереж, де пристрої розташовані у віддалених або важкодоступних місцях, де обслуговування та заміна батарей можуть бути ускладнені. NB-IoT також має високу проникність сигналу, що дозволяє забезпечити зв'язок у підвальних приміщеннях, густо забудованих міських районах та інших місцях з обмеженою видимістю.

Важливою перевагою NB-IoT є його здатність передавати невеликі обсяги даних з низькою швидкістю, що підходить для більшості IoT-застосувань, таких як моніторинг стану обладнання, управління освітленням, вимірювання рівня заповнення резервуарів тощо. Це робить NB-IoT відмінним вибором для різноманітних галузей, включаючи сільське господарство, логістику, розумні міста та промислові підприємства. Завдяки високій надійності, енергоефективності та широкому спектру застосувань, NB-IoT є однією з ключових технологій для розвитку Інтернету речей.

Sigfox

Sigfox — це глобальна мережа для Інтернету речей (IoT), яка забезпечує дуже низьку швидкість передачі даних з великою дальністю дії та низьким енергоспоживанням. Ця технологія призначена для передачі невеликих пакетів даних на великі відстані, що робить її ідеальною для додатків, де потрібно

відстежувати та моніторити різні параметри на віддалених об'єктах. Однією з основних особливостей Sigfox є її здатність працювати в нелицензованих діапазонах частот, що сприяє зниженню витрат на розгортання мережі.

Sigfox мережі відомі своєю простотою та ефективністю. Вони використовують ультра-вузькосмугову модуляцію (UNB), що дозволяє передавати дані на відстань до 10-40 км в сільській місцевості та до 3-10 км у міських умовах. Завдяки низькому енергоспоживанню пристрої можуть працювати від батареї до 10 років, що є важливою перевагою для IoT додатків, де часта заміна батарей є небажаною. Sigfox підтримує однонаправлену та двонаправлену передачу даних, що дозволяє використовувати цю технологію для різних типів сенсорних додатків.

Sigfox мережі забезпечують високу надійність та безпеку передачі даних. Кожен пристрій у мережі має унікальний ідентифікатор, а дані шифруються, щоб запобігти несанкціонованому доступу та забезпечити захист інформації. Використання хмарних сервісів для обробки та зберігання даних дозволяє інтегрувати Sigfox з іншими IoT платформами та додатками, що розширює можливості для аналізу та управління даними в реальному часі. Завдяки своїм характеристикам, Sigfox знаходить застосування у різних галузях, включаючи логістику, моніторинг інфраструктури, управління енергією та багато інших.

Архітектури

Сітчаста мережа (Mesh Network) — це тип мережі, в якій кожен пристрій (вузол) з'єднаний з кількома іншими вузлами, створюючи сітку зв'язків. Ця архітектура дозволяє даним передаватися через кілька шляхів від джерела до призначення, що забезпечує високу надійність і стійкість мережі. Якщо один з вузлів виходить з ладу, дані можуть автоматично перенаправлятися через інші вузли, що мінімізує ризик втрати зв'язку. Сітчасті мережі широко використовуються в додатках IoT, розумних будинках, промислових системах та міських інфраструктурах завдяки їхній гнучкості та масштабованості.

Безпека сітчастих мереж забезпечується через використання декількох рівнів захисту, таких як шифрування даних, автентифікація пристроїв та

контроль доступу. Шифрування забезпечує конфіденційність переданих даних, унеможливлюючи їх перехоплення та зчитування сторонніми особами. Автентифікація гарантує, що лише авторизовані пристрої можуть підключатися до мережі, знижуючи ризик несанкціонованого доступу. Крім того, контроль доступу дозволяє обмежити права різних вузлів у мережі, що забезпечує додатковий рівень захисту. Завдяки цим заходам, сітчасті мережі є надійними та безпечними для використання в різних критичних додатках.

Point-to-Point Architecture (Архітектура точка-точка)

Архітектура точка-точка (Point-to-Point) забезпечує прямий зв'язок між двома пристроями без проміжних вузлів або точок доступу. Це простий і ефективний спосіб передачі даних, який зазвичай використовується для встановлення надійного та швидкого з'єднання між двома віддаленими об'єктами. Архітектура точка-точка може бути використана в різних додатках, таких як з'єднання двох комп'ютерів, мережевих пристроїв або бездротових датчиків у системах моніторингу. Завдяки своїй простоті, вона забезпечує мінімальні затримки та високу швидкість передачі даних, що робить її ідеальною для критично важливих додатків, де необхідна висока пропускна здатність і надійність.

Безпека в архітектурі точка-точка забезпечується за допомогою шифрування даних та автентифікації пристроїв. Шифрування захищає передані дані від перехоплення та несанкціонованого доступу, забезпечуючи конфіденційність та цілісність інформації. Автентифікація гарантує, що обидва кінцеві пристрої є дійсними учасниками з'єднання, запобігаючи можливості підключення несанкціонованих пристроїв. Така архітектура може також використовувати додаткові заходи безпеки, такі як VPN (Virtual Private Network), для забезпечення захищеного каналу зв'язку навіть у відкритих мережах. Завдяки цим заходам, архітектура точка-точка забезпечує високий рівень безпеки та надійності передачі даних.

The Things Network (TTN)

Це відкрита платформа для створення LoRaWAN мереж, яка дозволяє

користувачам підключати IoT пристрої, обробляти дані та керувати ними. TTN надає інструменти та сервіси для створення масштабованих і безпечних IoT рішень, що робить її популярною серед розробників і підприємств по всьому світу.

Одна з основних переваг TTN — це можливість інтеграції з іншими хмарними сервісами та платформами. Користувачі можуть налаштовувати передачу даних з TTN на інші сервіси, такі як AWS IoT, Google Cloud IoT, Azure IoT Hub та інші. Це забезпечує гнучкість та розширює можливості для обробки та аналізу даних.

Безпека даних є критично важливою в будь-якій IoT мережі, і TTN забезпечує високий рівень захисту інформації. Всі дані, що передаються через LoRaWAN, шифруються за допомогою AES-128, що гарантує захист від перехоплення та несанкціонованого доступу. Крім того, TTN використовує механізми аутентифікації для перевірки пристроїв, що підключаються до мережі, забезпечуючи додатковий рівень безпеки.

Google Cloud IoT

Cloud IoT Core — це керований сервіс, який дозволяє безпечно підключати мільйони глобально розподілених пристроїв до Google Cloud. Він підтримує два основні протоколи для підключення пристроїв: MQTT та HTTP, що забезпечує гнучкість у виборі технології зв'язку. Cloud IoT Core забезпечує реєстрацію пристроїв, управління та моніторинг, а також шифрування даних для захисту інформації під час передачі.

Cloud Dataflow — це сервіс для обробки та аналізу потокових і пакетних даних у реальному часі. Він інтегрується з Cloud Pub/Sub, дозволяючи аналізувати дані з IoT пристроїв у режимі реального часу. Cloud Dataflow підтримує складні обчислювальні завдання, такі як агрегація, фільтрація та обробка подій.

Алгоритми

Mean Squared Error (MSE) – це алгоритм для оцінки якості зображень, що розраховує середнє квадратичне відхилення між відповідними пікселями двох

зображень. Спочатку обидва зображення мають однаковий розмір і формат. Потім обчислюється різниця між кожним відповідним пікселем, яка підноситься до квадрату. Середнє значення цих квадратів утворює MSE.

Переваги MSE в простоті, реалізації та швидкості обчислення, а недоліки в нечутливості до людського сприйняття та висока чутливість до зсувів та обертання зображень. MSE добре підходить для базової оцінки якості зображень, але в складніших випадках можуть знадобитися більш просунуті методи.

Normalized Cross-Correlation (NCC)

Normalized Cross-Correlation (NCC) – це метод для порівняння двох зображень або сигналів, який враховує їхню нормалізацію. NCC обчислює кореляцію між двома зображеннями, нормалізуючи значення пікселів, щоб зменшити вплив різниць у яскравості та контрасті. Спочатку обчислюється середнє значення та стандартне відхилення для кожного зображення. Потім обчислюється кореляція між нормалізованими значеннями пікселів двох зображень.

Переваги NCC включають стійкість до змін яскравості та контрасту, що робить його більш надійним для порівняння зображень у різних умовах освітлення. Однак NCC може бути менш ефективним у випадках значних геометричних трансформацій, таких як обертання або масштабування.

Perceptual Hashing

Perceptual Hashing – це метод для порівняння зображень, який створює компактні представлення зображень, відомі як хеші. Ці хеші генеруються на основі візуальних характеристик зображень, таких як яскравість і структура. Основна ідея полягає в тому, що схожі зображення матимуть схожі хеші, що дозволяє швидко порівнювати зображення.

Процес включає попередню обробку зображення (наприклад, зменшення розміру), перетворення його у градації сірого, розрахунок середнього значення яскравості пікселів, а потім побудову хешу на основі порівняння кожного пікселя зі середнім значенням.

Переваги методу Perceptual Hashing включають в собі високу швидкість та ефективність порівняння зображень, стійкість до незначних змін, таких як

зміни яскравості або розмиття. Однак цей метод може бути менш точним у випадках значних змін у зображеннях, таких як серйозне обертання або масштабування.

2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування

Для реалізації програми за темою бакалаврської дипломної роботи я вирішив скористатися інструментом MFC Visual Studio, який загалом полегшує створення нових додатків, використовуючи вже готові бібліотеки Microsoft Foundation Classes (MFC). Він автоматично генерує базову структуру проекту з усіма необхідними файлами і початковим кодом, що дозволяє розробникам зосередитися на додаванні специфічного функціоналу. Це значно зменшує час, витрачений на початкове налаштування проекту, забезпечуючи швидкий старт розробки.

MFC Visual Studio підтримує створення різних типів додатків, включаючи однодокументний інтерфейс (SDI), багатодокументний інтерфейс (MDI) та додатки з діалоговими вікнами. Це дозволяє вибрати оптимальний шаблон для конкретних потреб завдання, забезпечуючи відповідну структуру і початковий код. Крім того, інструмент пропонує налаштування різних параметрів проекту, таких як підтримка баз даних, наявність іконок, меню і панелей інструментів, що дозволяє створити проект, максимально відповідний вимогам додатку.

Використання Visual Studio забезпечує зручність роботи і підвищує продуктивність розробки мого программного забезпечення. MFC Visual Studio також підтримує використання сторонніх бібліотек і компонентів, що дозволяє розширити функціональні можливості додатку та інтегрувати його з іншими системами.

Завдяки використанню MFC Visual Studio, я зможу швидко створити професійний додаток з графічним інтерфейсом користувача, який відповідає

стандартам та рекомендаціям дипломної роботи за першим (бакалаврським) рівнем. Це забезпечує високу якість коду, легкість у підтримці та розширюваність проекту. MFC Visual Studio є потужним інструментом для створення складних Windows-додатків, що поєднує зручність, ефективність і надійність.

Звісно, для розробки програмного забезпечення треба доробити готовий код своїми знаннями по C++, але в мене наявні такі можливості. Цей інструмент MFC Visual Studio працює лише на цій програмній мові, тому прийшов час повністю викласти свої знання заради виконання практичної частини.

Якщо говорити взагалі про переваги та якості C++ то це потужна та високопродуктивна мова програмування, яка широко використовується для розробки програмного забезпечення, включаючи системи кібербезпеки, операційні системи, ігри, та інші додатки, де важлива продуктивність і ефективність використання ресурсів. C++ підтримує об'єктно-орієнтоване програмування (ООП), що сприяє модульності коду, повторному використанню та легкості підтримки.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, я повинен розробити програмне забезпечення, яке призначено для системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести огляд та аналіз існуючих систем кібербезпеки для захисту віддалених об'єктів. Документувати результати аналізу для врахування у подальших розробках;

б) обрати методику побудови системи кібербезпеки, яка найкраще

підходить для захисту віддаленого об'єкту з оповіщенням через бездротовий канал зв'язку. Обґрунтувати вибір методики на основі проведеного аналізу;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлене технічноме завдання. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) розробити зручний графічний інтерфейс користувача (GUI) за допомогою MFC, що дозволить легко налаштовувати систему та моніторити стан віддаленого об'єкту;

д) зробити висновки про створену роботу, яка буде уточнювати одержані результати та досвід роботи.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Система кібербезпеки для захисту віддалених об'єктів складається з кількох основних компонентів, які працюють у тісній взаємодії для забезпечення безперервного моніторингу, виявлення загроз та оперативного реагування. Основні компоненти системи включають сенсори, бездротові мережі зв'язку, центральний шлюз, сервер для обробки даних та пристрої для сповіщення відповідальних осіб.

Функціонування системи починається зі збору даних за допомогою сенсорів, встановлених на віддалених об'єктах. Ці сенсори здійснюють моніторинг різних параметрів, таких як температура, вологість, рівень освітлення, рух та вібрації. Сенсори обладнані мікропроцесорами, які здійснюють попередню обробку даних, що дозволяє зменшити обсяг даних, які передаються на центральний шлюз, та підвищити ефективність системи.

Зібрані дані передаються на центральний шлюз через бездротові мережі, використовуючи такі технології, як LoRaWAN, NB-IoT та Sigfox. Ці технології забезпечують стабільний зв'язок на великих відстанях та при мінімальному енергоспоживанні. Центральний шлюз отримує дані від різних сенсорів, об'єднує їх та передає на сервер для подальшої обробки.

Центральний сервер приймає дані від шлюзу та виконує їх аналіз за допомогою спеціалізованого програмного забезпечення. Це програмне забезпечення включає в себе алгоритми машинного навчання та штучного інтелекту, які дозволяють ефективно обробляти великі обсяги даних, виявляти аномалії та ідентифікувати потенційні загрози. Результати обробки використовуються для прийняття рішень в реальному часі, що дозволяє забезпечити високу оперативність та точність реакції на загрози.

У разі виявлення загрози система автоматично генерує сповіщення та

передає його відповідальним особам через різні канали зв'язку, такі як SMS, електронна пошта або спеціалізовані додатки. Сповіщення містять детальну інформацію про виявлену загрозу, можливі наслідки та рекомендації щодо реагування. Це дозволяє відповідальним особам оперативно вжити необхідних заходів для усунення проблеми.

Дальність дії систем та перешкоди

Бездротові технології мають різні характеристики щодо дальності дії та здатності проходити через перешкоди. Візьмемо наприклад LoRaWAN, який може передавати дані на відстань до 15 км у сільській місцевості та до 5 км у міських умовах, завдяки використанню технології модуляції LoRa. Це робить його ідеальним для використання на великих фермерських господарствах, у лісових районах або на промислових об'єктах, де необхідно покрити великі території з мінімальними витратами на енергоспоживання.

На пропускну здатність та стабільність бездротового зв'язку можуть бути вплинуті різні перешкоди, такі як фізичні об'єкти, погодні умови та електромагнітні завади. Наприклад, стіни, дерева та інші фізичні перешкоди можуть значно зменшувати дальність дії радіохвиль, особливо для високочастотних сигналів. Однак, низькочастотні сигнали мають кращу здатність проходити через перешкоди та можуть забезпечувати більш стабільний зв'язок у складних умовах. Електромагнітні завади, такі як сигнали від інших бездротових пристроїв або електричних приладів, можуть створювати інтерференцію, що також впливає на якість зв'язку. Використання ліцензованих спектрів може допомогти мінімізувати вплив таких завад, забезпечуючи більш стабільний та надійний зв'язок.

3.2 Розробка структурної схеми

Структурна схема системи зображена на рисунку 3.1.

На схемі зображена структура системи збору та обробки даних у мережі LoRaWAN. У верхній частині схеми знаходяться системи збору даних, які

складаються з сенсорів і шлюзів. Сенсори встановлені на віддалених об'єктах і збирають різноманітні дані, такі як температура, вологість, рух тощо. Зібрані дані передаються сенсорами на шлюзи через бездротовий зв'язок. Шлюзи працюють як проміжні пристрої, які отримують дані від кількох сенсорів та передають їх далі на обробку через IP-мережу (TCP/IP).

Далі дані надходять в інтернет. Інтернет забезпечує передачу даних від шлюзів до центральних серверів для подальшої обробки. Використання протоколу TCP/IP гарантує надійну і безпечну передачу даних на великі відстані, забезпечуючи доступ до мережі з будь-якої точки світу.

У нижній частині схеми розміщені панелі моніторингу та модуль автентифікації в рамках системи автентифікації користувача. Панелі моніторингу отримують дані з інтернету і відображають їх у вигляді графіків, діаграм та інших візуальних елементів, надаючи користувачам можливість аналізувати інформацію в реальному часі. Панелі моніторингу також можуть генерувати сповіщення у разі виявлення аномалій або загроз, що дозволяє оперативно реагувати на події.

Модуль автентифікації відповідає за перевірку прав доступу користувачів до системи. Він забезпечує, щоб тільки авторизовані користувачі могли отримати доступ до даних і панелей моніторингу. Це забезпечує додатковий рівень безпеки, захищаючи систему від несанкціонованого доступу та маніпуляцій з даними.

Таким чином, схема відображає потік даних від сенсорів через шлюзи і інтернет до панелей моніторингу та модуля автентифікації, забезпечуючи надійний і безпечний спосіб збору, передачі та обробки інформації у системі кібербезпеки для віддалених об'єктів.

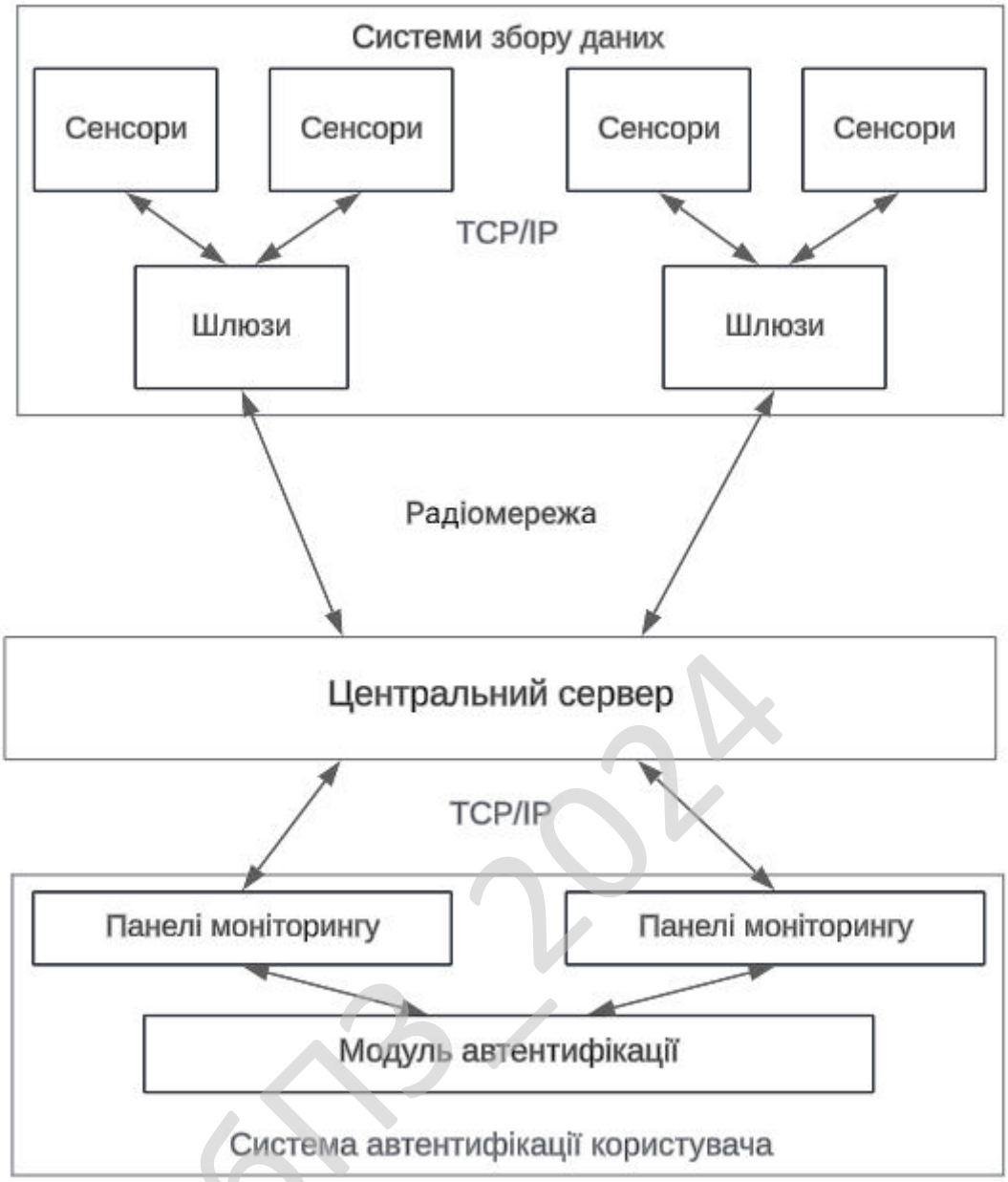

Рисунок 3.1 – Структурна схема системи

Збір даних є першим і ключовим етапом функціонування системи кібербезпеки для віддалених об'єктів. На цьому етапі використовуються кінцеві пристрої, оснащені різноманітними сенсорами, які встановлюються на об'єктах для моніторингу різних параметрів. Сенсори можуть бути різних типів залежно від потреб конкретної системи: температурні сенсори, сенсори вологості, датчики руху, газові детектори, сенсори тиску та багато інших. Кожен з цих сенсорів виконує специфічну функцію, збираючи дані про навколишнє середовище або стан обладнання.

Сенсори постійно або періодично зчитують дані і передають їх на кінцевий пристрій через TCP/IP по проводах. Кінцеві пристрої можуть об'єднувати кілька сенсорів, забезпечуючи збір комплексних даних з різних джерел. Збір даних також включає первинну обробку інформації на рівні кінцевого пристрою. Це може бути необхідно для фільтрації шуму, усереднення даних або виявлення базових аномалій. Наприклад, якщо сенсор руху виявляє постійний рух, кінцевий пристрій може здійснити додаткову перевірку для визначення типу руху та його джерела перед тим, як передати цю інформацію далі по системі.

Шлюзи отримують дані від сенсорів і передають їх на центральний сервер через радіомережу, використовуючи такі технології, як LoRaWAN, NB-IoT або Sigfox. Ці технології дозволяють забезпечити стабільний зв'язок на великих відстанях з мінімальним енергоспоживанням, що є критично важливим для віддалених об'єктів. Бездротові мережі дозволяють кінцевим пристроям передавати зібрані дані на значні відстані до найближчих шлюзів. Шлюзи виконують роль проміжних вузлів, які приймають дані від кількох кінцевих пристроїв одночасно.

Шлюзи також відповідають за забезпечення безпеки під час передачі даних на центральний сервер. Це включає використання криптографічних методів для шифрування даних, що передаються, а також аутентифікацію пристроїв, щоб гарантувати, що дані надходять тільки від авторизованих сенсорів. Ці заходи допомагають запобігти несанкціонованому доступу до даних та захищають систему від потенційних кібератак.

Центральний сервер приймає дані від шлюзів і розподіляє їх для подальшої обробки та зберігання. Виконується аналіз за допомогою спеціалізованого програмного забезпечення. Це програмне забезпечення включає в себе готові працюючі алгоритми, які дозволяють ефективно обробляти великі обсяги даних, виявляти аномалії та ідентифікувати потенційні загрози за потребою користувачів. Завдяки цим процесам, центральний сервер перетворює сирі дані, отримані від сенсорів, на корисну інформацію, яка може

бути використана для прийняття рішень, моніторингу та управління віддаленими об'єктами.

Після обробки даних сервер надсилає інформацію авторизованим користувачам через Інтернет за допомогою TCP/IP. Інформація може містити дані про час і місце виявлення руху, а також характер руху (дрібний або великий). Камери та сенсори, розташовані на віддаленому об'єкті, постійно збирають інформацію та відправляють її на сервер через мережу LoRaWAN. Ця мережа забезпечує надійну передачу даних на великі відстані з мінімальними витратами енергії.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Програмне забезпечення функціонально складається з наступних блоків, починаючи зі сторони сервера (лівої частини):

– Блок зберігання даних

– Блок конфігурації, запуску

– Блок призначення прав доступу

– Блок моніторингу подій

– Блок обробки зображень

– Блок захисних протоколів

– Мережевий блок

Та блоків зі сторони користувача (правої частини)

– Панелі моніторингу

– Модуль автентифікації

– Блок налаштувань користувача

– Блок сповіщень

– Блок звітності

Функціональна схема - це схема, яка описує суть роботи пристроїв,

програм і взаємодій користувачів з цим, та має більш узагальнений рівень, ніж структурний.

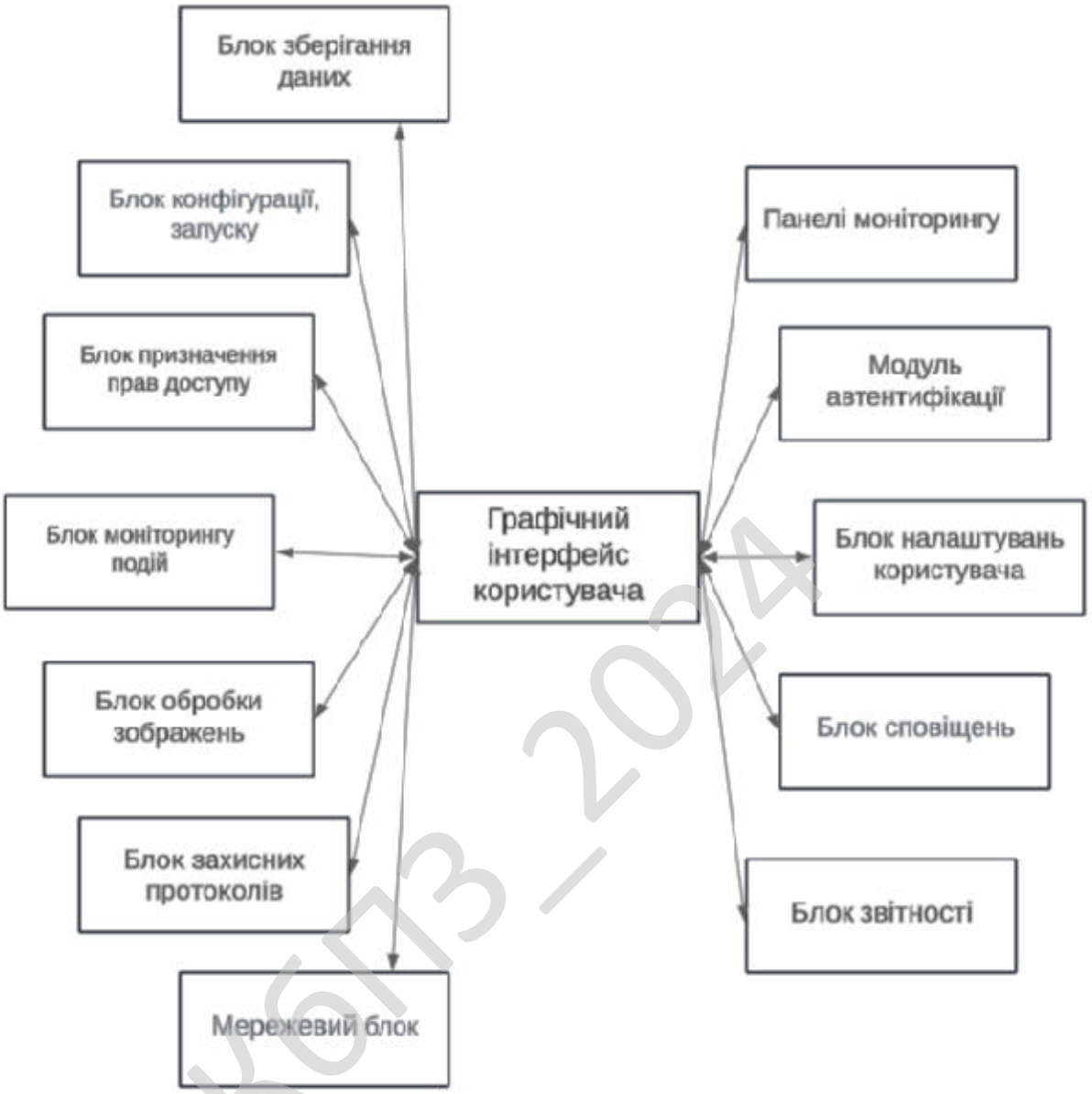

Рисунок 3.2 – Функціональна схема роботи системи

Графічний інтерфейс користувача (GUI) є центральним елементом системи, що забезпечує взаємодію користувача з різними функціональними блоками. Перш за все, через блок конфігурації та запуску користувач може налаштувати систему відповідно до своїх потреб і вимог. Це включає визначення параметрів роботи сенсорів та камер, а також налаштування способів сповіщення.

У цій функціональній схемі розглядається загальна взаємодія між клієнтом та сервером, яка стає доступною після вдалої авторизації.

Як зображено на функціональній схемі, на стороні клієнта формується модель, що складається з інформації про виявлені рухи навколо віддаленого об’єкту, після чого відправляється на сервер через захищену мережу, наприклад LoRaWAN.

Сервер обробляє отримані дані, перевіряє, та передає їх користувачам, надсилаючи усю потрібну інформацію, яка вже оброблюється системою. Ця система має обов’язок виводити на екрани користувачів інформацію, яка була налаштована заздалегідь, та відповідно до його налаштувань повинна повідомляти про зафіксовані рухи. З моменту реагування на події система почне запис відео, заради можливості аналізу зафіксованих рухів біля віддаленого об’єкту.

Завдання моніторингу, сповіщення та захисту віддаленного об’єкту поки що не вирішено у повному обсязі. Однак, у рамках істотних обмежень, є методи, що дозволяють наблизитися до реалізації программного забезпечення, здатного реалізувати систему захисту віддалених об’єктів, що знаходяться дуже далеко або ж у важкодоступних місцях.

Коли ми дивимося на сучасні системи кібербезпеки, ми часто не усвідомлюємо, який обсяг роботи проробляється для забезпечення безпеки віддалених об'єктів. Здавалося б, прості задачі, такі як виявлення несанкціонованого доступу або сповіщення про небезпеку, можуть бути легко автоматизовані. Проте, є кілька важливих аспектів, які ускладнюють цей процес, і які ми розглянемо нижче:

Масштаб системи. Віддалені об'єкти можуть мати різні розміри та конфігурації. Це вимагає налаштування сенсорів та камер для забезпечення оптимального покриття території.

Розташування об'єктів. Об'єкти, що потребують захисту, можуть бути розташовані в різних місцях з різною мережею доступу. Це вимагає використання різних методів передачі даних, таких як Wi-Fi, мобільні мережі

або супутниковий зв'язок.

Інтерференція та перешкоди. Системи безпеки можуть зустрічати численні перешкоди, такі як погана видимість, фізичні перешкоди, погодні умови або електромагнітні завади. Ці фактори можуть впливати на точність роботи сенсорів і камер.

Динамічні зміни в середовищі. Об'єкти можуть змінюватися з часом, наприклад, через будівельні роботи або зміни ландшафту. Система повинна адаптуватися до таких змін, забезпечуючи постійну надійність зв'язку та передачі отриманих даних.

Проекція та кути огляду. Камери та сенсори можуть сприймати об'єкти під різними кутами, що може впливати на точність виявлення. Поворот об'єкта або зміна кута огляду можуть значно змінити картину, що ускладнює автоматичне розпізнавання загроз, доводиться пильно налаштовувати придбане обладнання заради кращого використання.

Розглянемо конкретно кожний блок функціональної схеми системи по ліву сторону схеми, тобро зі сторони серверу:

- Блок конфігурації, запуску: Цей блок відповідає за первинне налаштування системи та її запуск. Користувач може встановлювати параметри роботи сенсорів, камер, а також налаштовувати способи отримання сповіщень. Блок забезпечує гнучкість налаштувань, що дозволяє адаптувати систему до специфічних вимог та умов експлуатації.

- Блок призначення прав доступу відповідає за керування правами користувачів у системі. Він забезпечує захист доступу, встановлюючи різні рівні привілеїв для різних користувачів, що дозволяє уникнути несанкціонованого доступу до важливих функцій та даних системи.

- Блок моніторингу подій: Цей блок відповідає за постійне спостереження та аналіз подій, що відбуваються в системі. Він відстежує всі активності, включаючи рухи, зміни в налаштуваннях та інші значущі події. Блок моніторингу подій забезпечує збір і обробку даних у режимі реального часу, що дозволяє швидко виявляти і реагувати на потенційні загрози або аномалії в

роботі системи, підтримуючи її ефективність та безпеку.

- Блок захисних протоколів: Відповідає за забезпечення безпеки передачі даних і комунікацій у системі. Використовує різні методи шифрування та аутентифікації для запобігання несанкціонованому доступу і захисту даних від потенційних кібератак, забезпечуючи надійність і конфіденційність інформації.

- Блок обробки зображень: Цей блок отримує дані з камер та аналізує їх з метою виявлення руху. Використовуючи алгоритми порівняння зображень, він виділяє ключові точки та аналізує зміни, що можуть свідчити про рух. Цей процес дозволяє класифікувати рух як дрібний (наприклад, рух тварини, рослинності) або великий (рух людини, автівки, тощо).

- Блок зберігання даних: Відповідає за зберігання всіх отриманих даних, включаючи інформацію про виявлені рухи та налаштування користувача. Блок забезпечує надійне зберігання даних для подальшого аналізу та доступу в разі потреби, підтримуючи безпеку та цілісність інформації.

- Мережевий блок: Забезпечує передачу даних між різними компонентами системи через захищені канали зв'язку. Використовуючи мережу LoRaWAN, цей блок забезпечує надійну передачу даних на великі відстані з мінімальними витратами енергії, підтримуючи безперервну та ефективну роботу системи, а саме головне – безпечну, завдяки шифрування AES-128 ключом.

Ознайомившись з функціональною схемою зі сторони сервера перейдемо до огляду схеми, яка виконується для потреб користувачів, переглянемо праві блоки функціональної схеми.

- Панелі моніторингу дозволяють користувачам в реальному часі відстежувати стан віддалених об'єктів. Вони відображають дані, отримані від сенсорів, та виявлені аномалії, що дозволяє оперативно реагувати на будь-які загрози. Панелі забезпечують повний огляд системи та можливість детального аналізу даних.

- Модуль автентифікації відповідає за управління доступом користувачів до системи. Він забезпечує безпечний вхід в систему, використовуючи сучасні методи автентифікації, такі як паролі, двофакторна аутентифікація та

біометричні дані. Це гарантує, що доступ до системи мають тільки авторизовані користувачі.

- Блок сповіщень: Блок зазвичай займається надсиланням сповіщень користувачам про виявлені рухи. Сповіщення можуть бути відправлені через різні канали, такі як SMS або робочу електронну пошту, в залежності від налаштувань користувача. Блок забезпечує своєчасне інформування користувачів про можливі загрози, дозволяючи їм швидко реагувати на події та активовувати протоколи захисту об'єкту.

- Блок налаштувань користувача дозволяє індивідуально налаштовувати параметри системи відповідно до потреб кожного користувача. Він включає можливість змінювати налаштування сповіщень, управління правами доступу та інші персоналізовані налаштування, що забезпечує гнучкість та зручність використання.

- Блок звітності надає користувачам можливість генерувати детальні звіти про роботу системи та її ефективність. Звіти можуть включати інформацію про виявлені загрози, статистику використання та інші важливі дані, що дозволяє аналізувати роботу системи та приймати обґрунтовані рішення для її покращення.

Розглянувши головні блоки функціональної схеми обох сторін перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання бакалаврського проекту, наведена на рисунку 3.3. Після початку роботи розробленого програмного забезпечення ми потрапляємо до головного вікна ПЗ.

Після відкриття головного вікна ПЗ користувач проходить через блок доступу до системи, який забезпечує ідентифікацію та перевірку прав доступу. Звідси через блок моніторингу подій можна контролювати систему в реальному часі, обробляючи отримані зображення та порівнюючи їх для виявлення

аномалій. Далі, блок конфігурації та запуску дозволяє користувачу налаштовувати параметри системи, керувати базою даних, а також призначати права доступу іншим користувачам.

Після успішної автентифікації блок доступу до системи перевіряє та надає користувачеві відповідні права доступу відповідно до його ролі чи рівня повноважень у системі. Це забезпечує належний розподіл обов'язків та обмежує доступ до конфіденційних даних або критично важливих функцій лише для авторизованих осіб.

Система сповіщень відповідальна за відправку повідомлень про виявлені події, що дозволяє користувачам оперативно реагувати на потенційні загрози. Мережевий блок забезпечує передачу даних між різними компонентами системи через захищені канали зв'язку, забезпечуючи надійність і безпеку передачі інформації, та при потребі налаштування мережі.

Блок захисних протоколів гарантує захист даних та комунікацій у системі, використовуючи різні методи шифрування та аутентифікації для запобігання несанкціонованому доступу у потрібні випадки. Через блок зберігання даних здійснюється надійне зберігання всієї інформації про виявлені події та налаштування користувачів.

Рисунок 3.3 – Діаграма взаємодії процесів

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 наведено блок-схему основної програми. Її робота складається з виконання наступних кроків:

- Ініціалізація та відкриття головного вікна ПЗ.
- Завантажити ресурси ПЗ?
- Завантаження ресурсів
- Завантаження еталонного зображення
- Запуск потоку для порівняння зображень
- Регулярне отримання нових зображень
- Отримане зображення відрізняється?
- Оповіщення користувача про знайдену аномалію
- Виконати захисні протоколи?
- Ініціалізація захисних протоколів
- Виведення результатів моніторингу
- Закриття програми?

На рисунку 4.2 наведено блок-схему підпрограми обробки зображень. Її робота складається з виконання наступних кроків:

- Зчитування нового зображення з камери
- Перетворення зображення у відтінки сірого
- Обчислення різниці між поточним та еталонним зображенням
- Пошук контурів на пороговому зображені
- Визначення наявності змін?
- Оповіщення користувача про зміни
- Запис інформації про зміни в базу даних

- Повернення до зчитування нового зображення

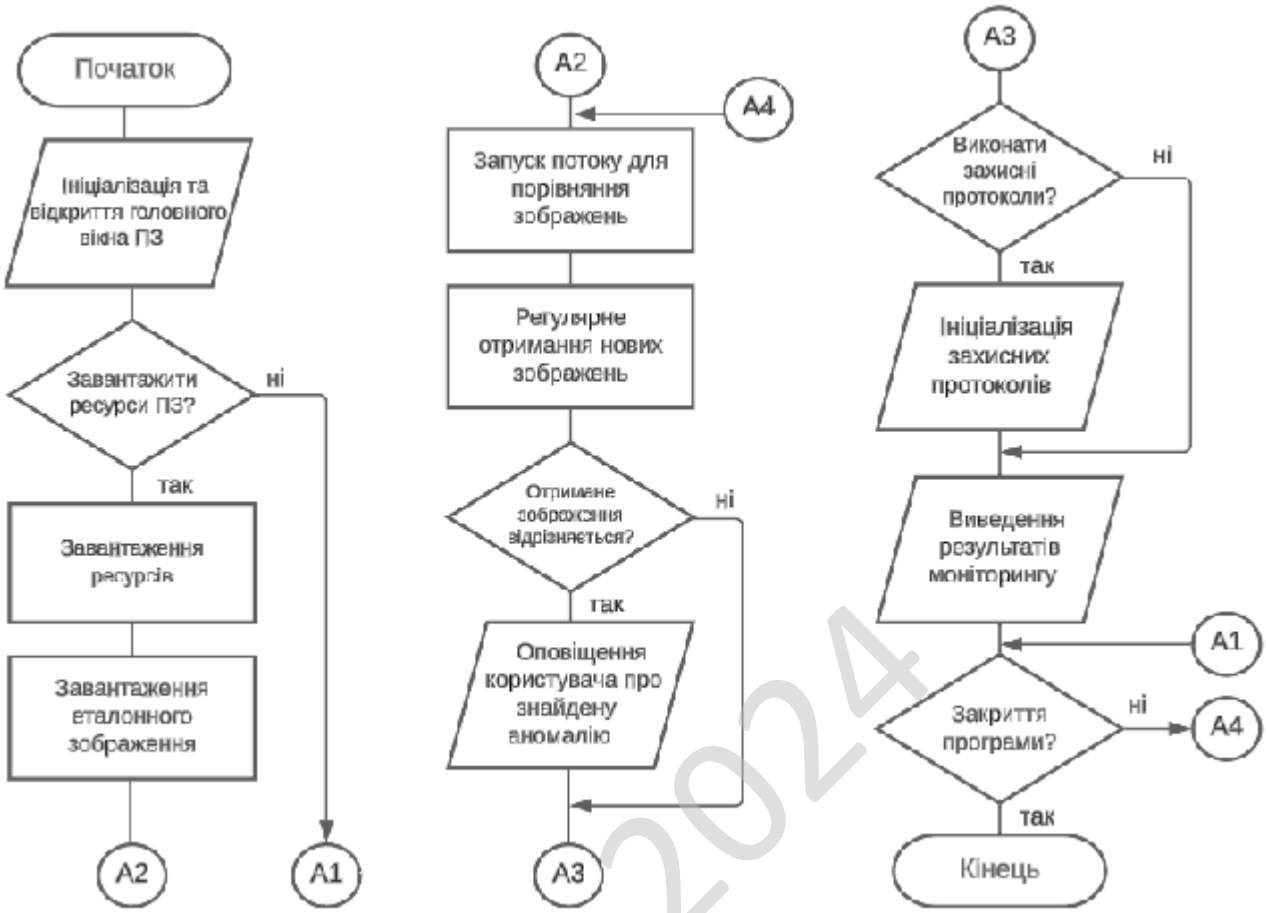

Рисунок 4.1 – Блок-схема основної програми

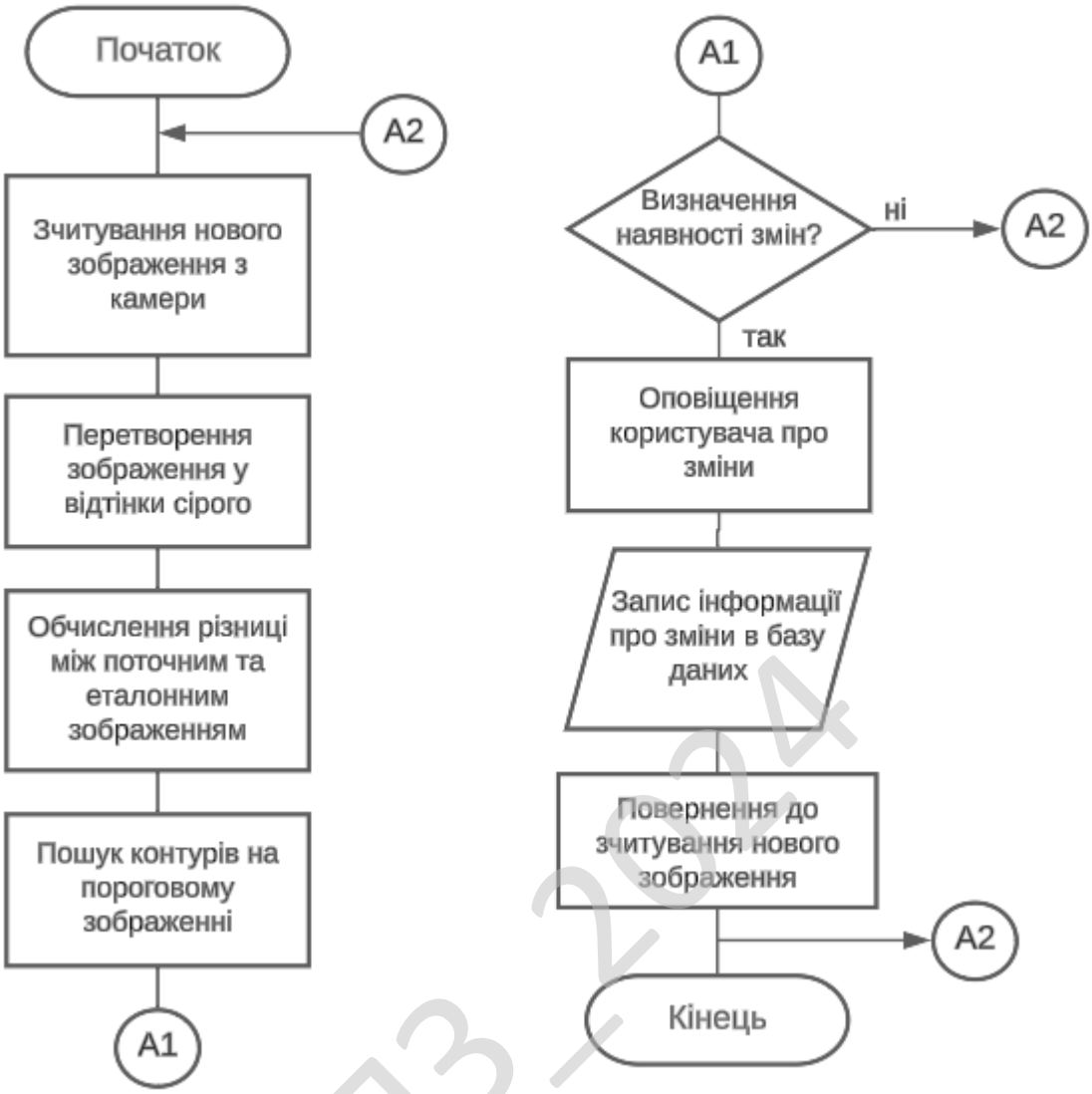

Рисунок 4.2 – Блок-схема підпрограми обробки зображень

Основна програма виконує завантаження необхідних ресурсів і еталонного зображення, після чого запускає підпрограму порівняння зображень, яка у свою чергу регулярно отримує нові зображення, порівнює їх з еталонним і, у разі виявлення аномалій, оповіщає користувача і виконує захисні протоколи.

Щодо вибіру алгоритма для реалізації свого технічного завдання для дипломної роботи я дослідив поширені алгоритми порівняння даних та обрав для себе алгоритм SSIM через його переваги та наближенність до реалізації моєї теми роботи.

Алгоритм SSIM (Structural Similarity Index Measure) є потужним інструментом для оцінки схожості між двома зображеннями. Він враховує три основні аспекти зображення: яскравість, контраст і структуру, що дозволяє більш точно оцінити якість і схожість зображень порівняно з традиційними методами. SSIM особливо корисний у задачах виявлення аномалій, оскільки дозволяє виявити навіть незначні відхилення в структурі зображень.

Опис алгоритму SSIM:

- Завантаження зображень: Програма отримує два зображення – поточне зображення та еталонне (зразкове) зображення.

- Попередня обробка: Зображення конвертуються в сірий колір, якщо вони кольорові, для спрощення подальшого аналізу.

- Вирівнювання розміру: Зображення масштабуються до однакового розміру, щоб порівняння було коректним.

- Розрахунок SSIM: Використовується структурне порівняння зображень (SSIM), яке оцінює схожість між двома зображеннями за кількома параметрами, такими як яскравість, контраст і структура.

- Аналіз результатів: Якщо значення SSIM нижче певного порогу, це означає, що на зображенні виявлено аномалію.

- Сповіщення користувача: Якщо виявлено аномалію, користувач отримує відповідне сповіщення.

SSIM має кілька переваг перед іншими алгоритмами для порівняння зображень. По-перше, він враховує людське сприйняття зображень, аналізуючи яскравість, контраст і структуру, що дозволяє більш точно оцінити схожість зображень з точки зору візуальної якості. Це робить SSIM більш чутливим до невеликих відхилень, які можуть бути важливими для виявлення аномалій. На відміну від простих методів, таких як MSE і PSNR, SSIM краще відображає суб'єктивну якість зображення, що робить його більш придатним для застосувань, де важлива візуальна точність.

4.2 Захист розробленого програмного забезпечення

Опис алгоритму використання AES-128 в системі LoRaWAN

Алгоритм AES-128 є центральним компонентом безпеки в LoRaWAN, який забезпечує шифрування даних і автентифікацію повідомлень, що передаються через мережу. Процес шифрування починається з генерації унікального 128-бітного ключа для кожного пристрою під час його активації. Цей ключ генерується за допомогою комбінації основного ключа (AppKey) і інформації про пристрій, такої як DevEUI (унікальний ідентифікатор пристрою) і JoinNonce (одноразовий номер, що генерується при кожній спробі приєднання).

Після активації пристрою, кожен пакет даних, що передається між пристроєм і шлюзом, шифрується за допомогою AES-128. Спочатку дані розбиваються на блоки по 16 байт. Якщо розмір даних не кратний 16 байтам, застосовується додаткове заповнення (padding), щоб досягти необхідної довжини. Кожен блок даних шифрується окремо, використовуючи 128-бітний ключ і алгоритм шифрування AES.

Процес шифрування включає кілька етапів:

а) Розширення ключа (Key Expansion): Як і під час шифрування, процес розшифрування починається з розширення початкового ключа (128 біт) до набору раундових ключів. Ці раундові ключі будуть використовуватися на різних етапах розшифрування.

б) Ініціалізація стану (State Initialization): Зашифрований блок (64 біти) поділяється на два 32-бітних підблоки (D0 і D1). Перший раундовий ключ застосовується до цих підблоків за допомогою операції XOR.

в) Раундові перетворення (Round Transformations): Кожен раунд включає три основні операції: додавання раундового ключа (AddRoundKey), змішування стовпців (MixColumns), перестановка рядків (ShiftRows) і підстановка байтів (SubBytes). Під час розшифрування ці операції виконуються в зворотному порядку:

- InvAddRoundKey: Додавання раундового ключа за допомогою операції XOR.

- InvMixColumns: Зворотна зміна стовпців.

- InvShiftRows: Зворотне зсування рядків.

- InvSubBytes: Зворотна підстановка байтів.

г) Фінальний раунд: Останній раунд не включає операцію змішування стовпців (MixColumns), але включає всі інші три операції.

Після завершення всіх раундів, отриманий результат є початковим незашифрованим текстом.

Заключний зашифрований блок передається через мережу LoRaWAN до шлюза, де він розшифровується з використанням того ж ключа. Процес дешифрування є зворотнім до процесу шифрування, включаючи інверсні операції для кожного з раундів.

Крім шифрування даних, AES-128 використовується для забезпечення цілісності повідомлень за допомогою алгоритму створення коду аутентифікації повідомлень (MIC). MIC додається до кожного пакета даних, що дозволяє шлюзу та серверу перевіряти, що дані не були змінені під час передачі.

Використання кодів MIC

Message Integrity Code (MIC) є важливим компонентом у забезпеченні безпеки передачі даних у мережах LoRaWAN. MIC представляє собою криптографічний код, який додається до кожного повідомлення, що передається через мережу, для забезпечення цілісності і автентичності даних.

MIC створюється за допомогою алгоритму AES-128 в режимі CMAC (Cipher-based Message Authentication Code). Під час цього процесу дані повідомлення обробляються разом із секретним ключем для генерування унікального коду. Цей код додається до повідомлення перед його відправленням. Отримувач, маючи доступ до того ж секретного ключа, може знову обчислити MIC і порівняти його з отриманим кодом. Якщо обчислений і отриманий MIC співпадають, це підтверджує, що повідомлення не було змінено під час передачі і походить від автентифікованого джерела.

Важливість MIC у LoRaWAN не можна переоцінити, оскільки він гарантує, що кожне повідомлення є автентичним і незмінним, забезпечуючи високу надійність і безпеку передачі даних у мережі. Використання MIC у поєднанні з шифруванням даних за допомогою AES-128 робить LoRaWAN стійкою до багатьох відомих атак, таких як атаки повтору та маніпуляції з даними.

Методи аналізу

Для забезпечення безпеки моєї системи, розробленої у дипломній роботі, актуальними є кілька методів аналізу, які можна застосувати для оцінки надійності шифрування та виявлення можливих вразливостей. Давайте розглянемо два більш актуальні ключові методи:

- Диференційний криптоаналіз є одним з найпоширеніших методів аналізу блочних шифрів. Він вивчає, як різниці у вхідних даних впливають на різниці у вихідних даних. Це може бути корисним для аналізу безпеки твоєї системи, оскільки дозволяє виявити можливі слабкі місця в алгоритмі шифрування. Однак, AES-128 був розроблений з урахуванням захисту від таких атак, тому цей метод буде корисним для підтвердження стійкості системи.

- Лінійний криптоаналіз шукає лінійні залежності між відкритим текстом, зашифрованим текстом і ключами. Він може бути використаний для виявлення вразливостей у схемі шифрування, які можуть бути неочевидними при використанні інших методів. Використання цього методу допоможе перевірити, чи є в твоїй системі приховані лінійні залежності, які можуть бути використані для зламу.

Висновок

Таким чином, була докладно описана структура алгоритму шифрування AES-128 і розглянуті методи його аналізу. Розглянуто принцип роботи AES-128, включаючи процеси шифрування та розшифрування, а також використання MIC для забезпечення цілісності та автентичності повідомлень у системах LoRaWAN. Найбільш прагматичними напрямками дослідження були визначені диференційний та лінійний криптоаналіз для оцінки стійкості алгоритму, атаки

на основі бокових каналів для виявлення вразливостей, пов'язаних з фізичними аспектами реалізації, а також захист від атак повторного використання.

Далі має бути створення програмної реалізації для більш детального розгляду алгоритму AES-128 і MIC, а також набір статистичних даних для повного дослідження і пошуку оптимального підходу до аналізу і забезпечення безпеки системи.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми. З нього видно, що інтерфейс користувача програми складається з таких логічних блоків:

- Вікна результату роботи сенсору, камери;
- Вікна виведення поточної інформації;
- Функціональної кнопки показу даних за останні 24 години;
- Функціональної кнопки налаштувань для користувача;
- Функціональної кнопки про дані розробника;
- Функціональної кнопки користування захисними протоколами.

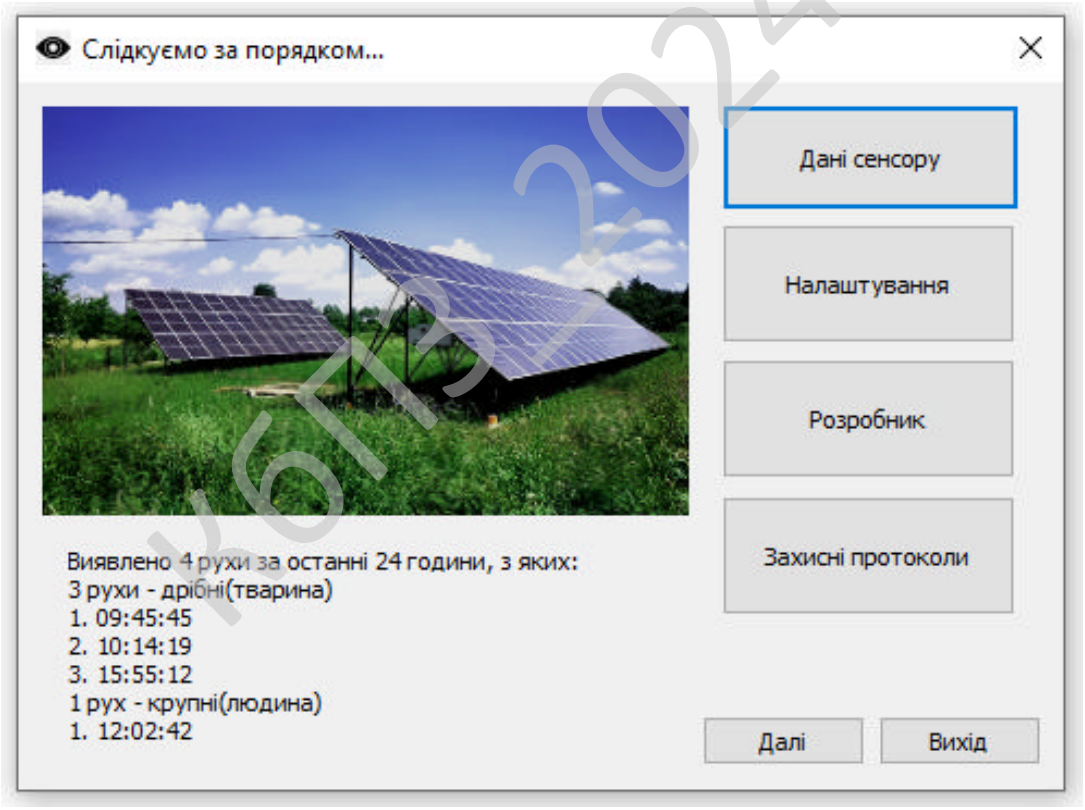

Рисунок 5.1 – Головне вікно програми

На рисунку 5.2 зображено форму авторського права. Для моєї дипломної роботи було обрано умови розповсюдження на основі моделі Shareware.

Користувачі зможуть завантажити і використовувати розроблене програмне забезпечення безкоштовно протягом певного пробного періоду. Після завершення пробного періоду користувачі, які бажають продовжити використання ПЗ, повинні будуть сплатити за ліцензію. Такий підхід дозволяє потенційним користувачам ознайомитися з функціоналом та перевагами вашого програмного забезпечення перед тим, як здійснити покупку, що може підвищити його популярність і комерційний успіх. Так само завдяки таким умовам розповсюдження потенційні покупці зможуть зрозуміти для себе корисна їм розроблена система або ж ні. Протягом певного часу, загалом два тижні, користувачі, що зареєструвалися на сайті розробленного додатку, можуть оцінити для себе використання цієї системи, Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його, заплативши авторові певну суму. В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

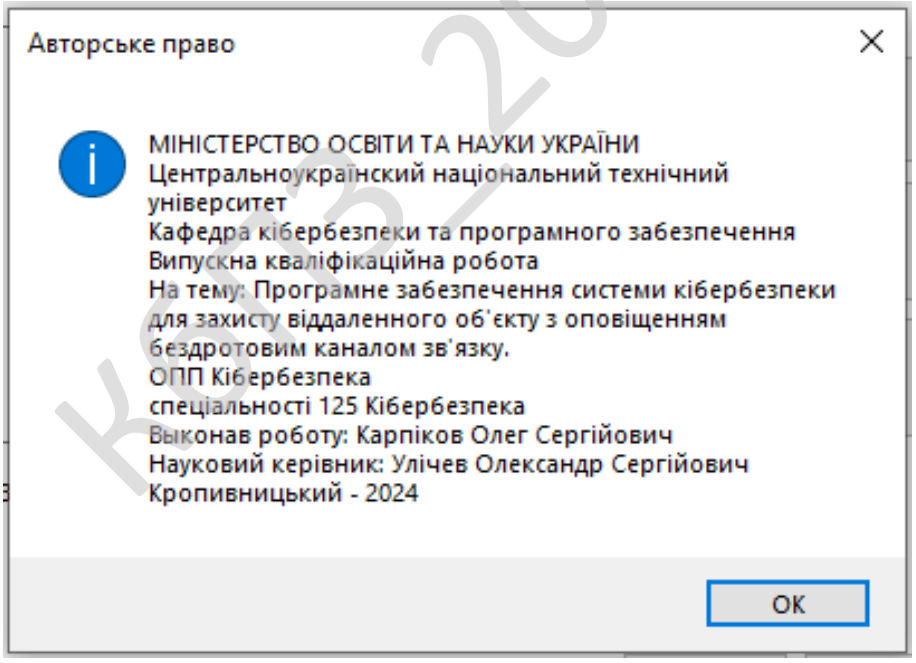

Рисунок 5.2 – Авторське право

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку.

В межах України налічується декілька компаній які дозволяють майбутнім користувачам користуватися їх послугами заради встановлювання технологій для захисту віддалених об'єктів.

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку.

– Досліджена система для захисту віддаленого об'єкту, що охоплює такі різновиди як сонячні батареї або інші критичні структури завдяки бездротовим технологіям.

– На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання для отримання інформації, шифрування, її безпечної передачі по мережі до користувача та подальше розшифрування.

Розроблене програмне забезпечення володіє простим, зрозумілим та дружнім, для користувача, інтерфейсом, який забезпечує простоту використання, легкість освоєння роботи з програмними продуктами і не вимагає особливих спеціальних знань.

При створені програмного забезпечення було використано об’єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма призначена для виконання під управлінням багатозадачної операційної системи кібербезпеки Windows 10/11.

Ця програма написана мовою C++ з допомогою Visual C++ (Microsoft Visual Studio) як середовища розробки та MFC (Microsoft Foundation Class) бібліотеки до створення GUI.

Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати часу на його розробку.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм AES-128 та MIC.

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

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мельник, А. М., & Берестенко, Д. О. (2022). ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ БЕЗДРОТОВИХ КАНАЛІВ ЗВ'ЯЗКУ. Тези доповідей, 44.

2. Антонова, Г. В., & Ковирьова, О. В. (2018). Бездротові технології як ланка цифровізації сільського господарства. Комп'ютерні засоби, мережі та системи, (17), 53-59.

3. Киричек, Г. Г., & Киричек, Г. Г. (2019). Робоча програма навчальної дисципліни" Бездротові технології".

4. Панченко, С. А. (2020). Бездротові мережі зв'язку для керування елементами приміщення: принцип роботи та параметри (Master's thesis, Сумський державний університет).

5. Карабут, А. О. (2021). Дослідження продуктивності безпроводових систем зв'язку.

6. Суровцев, О. А., & Павловський, О. М. (2018). Порівняння сучасних бездротових технологій обміну інформацією.

7. Криворучко, І. П. Аналіз варіантів реалізації бездротового зв'язку при створенні засобів технічної діагностики енергетичного обладнання.

8. Макаренко, А. С., Парфенова, А. А., & Могильный, С. Б. (2010). Бездротові технології передачі даних Wi-Fi, Bluetooth та ZigBee. Вісник Національного технічного університету України Київський політехнічний інститут. Серія: Радіотехніка. Радіоапаратобудування, (41), 171-181.

9. Родін, С. О. (2013). Технології зв'язку в системах охорони. Види та особливості застосування. Вісник Національного університету Львівська політехніка. Автоматика, вимірювання та керування, (774), 116-119.

10. СТЕПАНОВ, М., & ЛАВРІНЕНКО, В. (2023). ВИБІР ТЕХНОЛОГІЇ ЗВ'ЯЗКУ ЕЛЕМЕНТІВ СЕНСОРНОЇ МЕРЕЖІ. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES, (3), 248-255.

11. Гурик, О. Р. (2018). Обгрунтування структури безпровідних систем зв’язку для Smart-технологій (Master's thesis).

12. Райтер, П. М., & Григоришин, О. М. (2017). Аналіз переваг та недоліків сучасних технологій передачі даних для розподілених систем технічної діагностики і моніторингу.

13. Пархоменко, В. Л., Щепак, А. С., & Пархоменко, В. В. (2024). Моделювання та вдосконалення цифрових засобів обміну інформацією. Наукові записки Державного університету інформаційно-комунікаційних технологій, (1), 105-111.

14. Данилюк, Ю. Б. (2018). Технології передавання даних в комп’ютерних мережах (Master's thesis).

15. LoRaWAN, T. M. (2017). LoRaWAN. Beaverton, OR, USA.

16. Рибак, О. О. (2020). ДОСЛІДЖЕННЯ ЗАВАДОСТІЙКОСТІ ТЕХНОЛОГІЇ LORAWAN. Збірник матеріалів Міжнародної науково-технічної конференції «ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ».

17. Подорожняк, А. О., & Давиденко, А. О. (2021). Дослідження застосування технології LoRaWAN в аграрній галузі (Doctoral dissertation, ФОП Тарасенко ВП).

18. Шельпяков, В. (2021). Енергоефективність мережевої технології LoRaWAN в інфраструктурі сучасного міста. Відновлювана енергетика та енергоефективність у XXI столітті: матеріали XXII міжнародної науково-практичної конференції (Київ, 20-21 травня 2021 р.).

19. Слободюк, В. М. (2019). Дослідження потенційних можливостей технології lorawan в умовах міської забудови.

20. Самолюк, І. А., & Войцеховська, О. В. (2018). Перспективи розвитку бездротових телекомунікаційних технологій (Doctoral dissertation, ВНТУ).

21. Курєєв, А., & Банків, Д. (2017). Аналіз ефективності гетерогенних мереж Wi-Fi HaLow. In ITiC 2017 (pp. 406-414).

22. Tian, L., Santi, S., Seferagić, A., Lan, J., & Famaey, J. (2021). Wi-Fi HaLow for the Internet of Things: An up-to-date survey on IEEE 802.11 ah research. Journal of Network and Computer Applications, 182, 103036.

23. Гончаренко, Д. В., Мокін, В. Б., & Проценко, Д. П. (2023). Переваги технологій інтернету речей SIGFOX для створення локальної системи моніторингу атмосферного повітря (Doctoral dissertation, ВНТУ).

24. Гончаренко, Д. В., Мокін, В. Б., Проценко, Д. П., Горячев, Г. В., & Варчук, І. В. (2024). Іот-система вимірювання параметрів стану вод на базі sigfox (Doctoral dissertation, ВНТУ).

25. Проценко, Д. П., Цвігун, С. А., & Гончаренко, Д. В. (2023). Аналіз зони покриття станції інтернету речей sigfox для визначення місць розташування датчиків (Doctoral dissertation, ВНТУ).

26. Бердник, Ю. В. (2019). Можливості NB-IoT для захищеного зв'язку.

27. Сеник, А. О. (2020). *Використання технології NB-IoT для побудови сенсорних мереж* (Bachelor's thesis, КПІ ім. Ігоря Сікорського).

28. Сеник, А. О. (2021). Особливості використання технології NB-IoT для застосувань Інтернету речей (Master's thesis, КПІ ім. Ігоря Сікорського).

29. Хандогин, В. Д. (2022). Передача данных через протокол TCP IP. Системный администратор, (1-2), 230-231.

30. Колесник, А. О. (2016). Засіб криптографіно захищеного зв'язку на базі AES-128 (Rijndael).

31. Tsai, K. L., Huang, Y. L., Leu, F. Y., You, I., Huang, Y. L., & Tsai, C. H. (2018). AES-128 based secure low power communication for LoRaWAN IoT environments. Ieee Access, 6, 45325-45334.

32. Трунов, О. М. (2023). Датчики та сенсори робототехнічних систем.

33. Білан, С. М., & Шварц, І. М. (2005). Диференціальний криптоаналіз блоково-динамічного алгоритму шифрування.

34. Казміревський, В. В. (2022). Дослідження стійкості до лінійного та диференційного криптоаналізу функцій гешування (Doctoral dissertation, ВНТУ).

35. Козак, Р., & Прошин, С. (2011). Напрямки розвитку криптоаналізу. Збірник тез доповідей XV наукової конференції Тернопільського національного технічного університету імені Івана Пулюя, 76-76.

36. Авраменко, В. В., & Скаковська, А. М. (2015). Програмування на Visual C++ із застосуванням бібліотеки MFC.

37. Rehman, A., Wang, Z., Brunet, D., & Vrscay, E. R. (2011, May). SSIM-inspired image denoising using sparse representations. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1121-1124). IEEE.

38. Nilsson, J., & Akenine-Möller, T. (2020). Understanding ssim. arXiv preprint arXiv:2006.13846.

39. Пилявський, В. В., Пилявский, В. В., Патлаєнко, М. О., Патлаенко, М. О., & Таран, А. П. (2020). СИСТЕМА МОНІТОРИНГУ НА БАЗІ ТЕХНОЛОГІЇ LoRaWAN.

40. Крикун, Є. О. (2020). Технологія побудови ефективної безпроводової мережі з використанням протоколу LoRaWAN.

41. Міночкін, Д. А., & Рибак, О. О. (2019). Аналіз безпеки інтернету речей за технологією LoRaWAN.

42. Коваленко, С. М. (2018). Використання технології LoRa для радіозв'язку.

43. Diachuk, O. V. (2020). Порівняльний аналіз технологій класу LPWAN. Електронна та Акустична Інженерія, 3(3), 40-44.

44. Курілов, М. С., Курилов, М. С., Шмельова, Т. Р., & Шмелёва, Т. Р. (2020). АНАЛІЗ ЕФЕКТИВНОСТІ ТЕХНОЛОГІЇ NB-IoT МЕРЕЖІ LPWAN.

45. Болінова, М. М. (2019). Дослідження захищеності LPWAN мереж.

46. Лукащик, В., & Гріненко, Т. О. (2019). Автентифікація користувачів у системах Інтернету речей.

47. Морозов, Ю. В., & Пастернак, І. І. (2011). Класифікація засобів модульної взаємодії між клієнтом і сервером.

48. Кулик, І. А., & Зінченко, І. С. (2016). Технології передачі даних по бездротових сенсорних мережах (Doctoral dissertation, Сумський державний університет).

49. Шевченко, В. С. ПОРІВНЮВАЛЬНИЙ АНАЛІЗ ПЕРСПЕКТИВНОСТІ БЕЗДРОТОВИХ ТЕХНОЛОГІЙ ПЕРЕДАЧІ ДАНИХ. VII-й НАУКОВО-ПРАКТИЧНИЙ СЕМІНАР, 226.

50. Мельник, А. М., & Берестенко, Д. О. (2022). ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ БЕЗДРОТОВИХ КАНАЛІВ ЗВ'ЯЗКУ. Тези доповідей, 44.

Додаток А
(обов’язковий)

Технічне завдання

Зміст

1 Найменування та область застосування

Це технічне завдання розповсюджується на системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку.

2 Підстава для розробки

Підставою для розробки служить завдання на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 135-02 від 01.04.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є цілком дослідницька та стосовна до теми література та її існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

– вибір та обґрунтування методів реалізації проекту;

– розробка програмного забезпечення системи та налаштування взаємодії системи кібербезпеки з операційною системою і користувачем;

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

– системи кібербезпеки для захисту віддалених об'єктів шляхом аналізу зображень та виявлення аномалій через сенсори.

– цілісність даних у процесі роботи та зберігання;

– простий, інтуїтивно зрозумілий інтерфейс для користувача.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмний модуль визначається всіма правилами, що відносяться до викликів стандартних процедур, функцій, методів і форм, які визначені в технічній документації середовища розробки.

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

– відносна вологість повітря до 75%;

– температура повітря: 18-21 град. по Цельсію;

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на персональних комп'ютерах, працювати в ОС Windows 10/11 та бути сумісним з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп’ютер Intel® Celeron/8 Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Visual C++.

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у вигляді опису структури даних, схем та опису алгоритмів, а також текстів вихідних модулів програмного забезпечення згідно з вимогами ЄСПД.

7 Перелік документів, що розробляються

– Структурна схема системи – 1 аркуш.
– Функціональна схема системи – 1 аркуш.
– Діаграма процесів – 1 аркуш.
– Блок-схема алгоритму роботи програми – 2 аркуша.
– Пояснювальна записка – 50 аркушів.

8 Етапи розробки

8.1 Збір та обробка інформації по темі випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти. Формулювання задачі для виконання випускної кваліфікаційної роботи (складання технічного завдання).

8.2 Проведення досліджень або експериментів для уточнення основних положень випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.

8.3 Розробка функціональних схем та блок-схем алгоритмів роботи програмного забезпечення.

8.4 Створення прототипу ПЗ.

8.5 Тестування програмного забезпечення та аналіз отриманих результатів.

8.6 Оформлення пояснювальної записки та виконання робіт по графічній частині.

9 Порядок контролю та приймання

9.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 19.05.2024 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 03.06.2024 р.

Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

_____ Улічев О.С.

Програмне забезпечення системи кібербезпеки для захисту віддаленого об'єкту з оповіщенням бездротовим каналом зв'язку

Лістинг програми

Код документу 12

Носій: USB-флеш-накопичувач

Загальна кількість аркушів: 18

Літера: РП

Кропивницький – 2024 року

Diplom_eyeDlg.cpp – Головний файл

```
/************************************************************
* --- Програма системи ідентифікації на основі розпізнавання відбитку долоні
користувача--- *
* *
************************************************************/

#include "pch.h"
#include "framework.h"
#include "Diplom_eye.h"
#include "Diplom_eyeDlg.h"
#include "afxdialogex.h"
#include "Resource.h"
#include <opencv2/opencv.hpp>
#include <thread>
#include <chrono>
#include <gdiplus.h>
#include "ProtocolsDlg.h"
#include "SettingsDlg.h"
#include "Diplom_eyeDlg.h"

using namespace Gdiplus;
using namespace cv;
using namespace std;

void detectMotion(const cv::Mat& refImage, const cv::Mat& currentImage, CString&
result);

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_ABOUTBOX };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);

protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(IDD_ABOUTBOX)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()

CMFCDiplomDlg::CMFCDiplomDlg(CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_MFC_DIPLOM_DIALOG, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    GdiplusStartup(&gdiplusToken, &gdiplusStartupInput, NULL);
}

CMFCDiplomDlg::~CMFCDiplomDlg()
{
```

```
    GdiplusShutdown(gdiplusToken);
}

void CMFCDiplomDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_STATIC_PICTURE, m_pictureControl);
    DDX_Control(pDX, IDC_STATIC_INFO, m_infoControl);
}

BEGIN_MESSAGE_MAP(CMFCDiplomDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BUTTON_SENSOR, &CMFCDiplomDlg::OnBnClickedButtonSensor)
    ON_BN_CLICKED(IDC_BUTTON_SETTINGS,
&CMFCDiplomDlg::OnBnClickedButtonSettings)
    ON_BN_CLICKED(IDC_BUTTON_DEVELOPER,
&CMFCDiplomDlg::OnBnClickedButtonDeveloper)
    ON_BN_CLICKED(IDC_BUTTON_PROTOCOL,
&CMFCDiplomDlg::OnBnClickedButtonProtocol)
END_MESSAGE_MAP()

BOOL CMFCDiplomDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // Додається перевірка ідентифікаторів та ініціалізацію елементів керування
    if (!m_pictureControl.SubclassDlgItem(IDC_STATIC_PICTURE, this))
    {
        AfxMessageBox(_T("Не вдалося ініціалізувати контроль зображення!"));
        return FALSE;
    }

    if (!m_infoControl.SubclassDlgItem(IDC_STATIC_INFO, this))
    {
        AfxMessageBox(_T("Не вдалося ініціалізувати контроль інформації!"));
        return FALSE;
    }

    // Додаткова ініціалізація
    LoadReferenceImage();
    StartComparison();

    return TRUE;  // Повертає TRUE, якщо фокус не встановлено на елемент
керування
}

void CMFCDiplomDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

void CMFCDiplomDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this);
        SendMessage(WM_ICONERASEBKGND,
reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);
        int cxIcon = GetSystemMetrics(SM_CXICON);
```

```
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialogEx::OnPaint();
    }
}

HCURSOR CMFCDiplomDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

void CMFCDiplomDlg::OnBnClickedButtonSensor()
{
    StartComparison();
}

void CMFCDiplomDlg::OnBnClickedButtonSettings()
{
    CSettingsDlg settingsDlg;
    settingsDlg.DoModal();
}

void CMFCDiplomDlg::OnBnClickedButtonProtocol()
{
    CProtocolsDlg protocolsDlg;
    protocolsDlg.m_protocolsList.AddString(_T("Протокол аутентифікації
користувачів: Забезпечує перевірку справжності користувачів перед наданням
доступу до системи."));
    protocolsDlg.m_protocolsList.AddString(_T("Протокол шифрування даних:
Забезпечує захист конфіденційності даних шляхом їх шифрування під час передачі
та зберігання."));
    protocolsDlg.m_protocolsList.AddString(_T("Протокол контролю доступу:
Визначає правила доступу користувачів до ресурсів системи."));
    protocolsDlg.m_protocolsList.AddString(_T("Протокол виявлення вторгнень:
Призначений для виявлення підозрілої активності та вторгнень у систему."));
    protocolsDlg.m_protocolsList.AddString(_T("Протокол резервного копіювання
даних: Забезпечує збереження резервних копій даних для їх відновлення у разі
втрати або пошкодження."));
    protocolsDlg.m_protocolsList.AddString(_T("Протокол відстеження активності:
Забезпечує моніторинг дій користувачів та системи для виявлення аномалій та
підозрілої активності."));
    protocolsDlg.m_protocolsList.AddString(_T("Протокол захисту від шкідливого
програмного забезпечення: Забезпечує виявлення та усунення шкідливого
програмного забезпечення у системі."));
    protocolsDlg.DoModal();
}

void CMFCDiplomDlg::OnBnClickedButtonDeveloper()
{
    CAboutDlg dlgAbout;

    SetWindowText(_T("Авторське право"));

    CString developerInfo;
    developerInfo.Format(_T("МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ\n")
        _T("Центральноукраїнский національний технічний університет\n")
        _T("Кафедра кібербезпеки та програмного забезпечення\n")
        _T("Випускна кваліфікаційна робота\n")
        _T("На тему: Програмне забезпечення системи кібербезпеки\n")
        _T("для захисту віддаленного об'єкту з оповіщенням\n")
        _T("бездротовим каналом зв'язку.\n")
        _T("ОПП Кібербезпека\n")
```

```
        _T("спеціальності 125 Кібербезпека\n")
        _T("Виконав роботу: Карпіков Олег Сергійович\n")
        _T("Науковий керівник: Улічев Олександр Сергійович\n")
        _T("Кропивницький - 2024"));

    MessageBox(developerInfo, _T("Авторське право"), MB_OK |
MB_ICONINFORMATION);
}


void CMFCDiplomDlg::LoadReferenceImage()
{
    CStringA
refImagePath("C:\\Users\\karpi\\Desktop\\MFC_Diplom\\reference.jpg");
    m_refImage = imread(refImagePath.GetString());
    if (m_refImage.empty())
    {
        AfxMessageBox(_T("Не вдалося завантажити еталонне зображення!"));
    }
}

void CMFCDiplomDlg::StartComparison()
{
    m_running = true;
    m_thread = std::thread([this]() {
        while (m_running)
        {
            CompareImages();
            std::this_thread::sleep_for(std::chrono::seconds(10));
        }
        });
}

void CMFCDiplomDlg::StopComparison()
{
    m_running = false;
    if (m_thread.joinable())
    {
        m_thread.join();
    }
}

void CMFCDiplomDlg::CompareImages()
{
    CStringA
currentImagePath("C:\\Users\\karpi\\Desktop\\MFC_Diplom\\sens_image.bmp");
    Mat currentImage = imread(currentImagePath.GetString());

    if (currentImage.empty())
    {
        AfxMessageBox(_T("Не вдалося завантажити поточне зображення!"));
        return;
    }

    Mat grayRef, grayCurrent, diffImage, threshImage;
    cvtColor(m_refImage, grayRef, COLOR_BGR2GRAY);
    cvtColor(currentImage, grayCurrent, COLOR_BGR2GRAY);

    absdiff(grayRef, grayCurrent, diffImage);
    threshold(diffImage, threshImage, 25, 255, THRESH_BINARY);

    std::vector<std::vector<cv::Point>> contours;
    findContours(threshImage, contours, RETR_EXTERNAL, CHAIN_APPROX_SIMPLE);

    CString result;
    if (!contours.empty())
    {
        result = _T("Виявлено рух!");
    }
```

```
    else
    {
        result = _T("Рух не виявлено.");
    }

    m_infoControl.SetWindowText(result);
}

void CMFCDiplomDlg::LoadAndDisplayImage(CString filePath)
{
    CRect rect;
    m_pictureControl.GetClientRect(&rect);

    Graphics graphics(m_pictureControl.GetDC()->GetSafeHdc());
    Image image(filePath);
    graphics.DrawImage(&image, 0, 0, rect.Width(), rect.Height());
    m_pictureControl.ReleaseDC(m_pictureControl.GetDC());
}

void detectMotion(const cv::Mat& refImage, const cv::Mat& currentImage, CString&
result)
{
    cv::Mat grayRef, grayCurrent, diffImage, threshImage;

    cv::cvtColor(refImage, grayRef, cv::COLOR_BGR2GRAY);
    cv::cvtColor(currentImage, grayCurrent, cv::COLOR_BGR2GRAY);

    cv::absdiff(grayRef, grayCurrent, diffImage);

    cv::threshold(diffImage, threshImage, 25, 255, cv::THRESH_BINARY);

    std::vector<std::vector<cv::Point>> contours;
    cv::findContours(threshImage, contours, cv::RETR_EXTERNAL,
cv::CHAIN_APPROX_SIMPLE);

    if (contours.size() > 0)
    {
        result = CString(_T("Виявлено рух!"));
    }
    else
    {
        result = CString(_T("Рух не виявлено."));
    }
}

// Завантаження еталонного зображення
void CMFCDiplomDlg::LoadReferenceImage()
{
    CStringA refImagePath(R"(E:\\MFC_Diplom\\reference.jpg)");
    m_refImage = cv::imread(refImagePath.GetString());

    if (m_refImage.empty())
    {
        AfxMessageBox(_T("Не удалось загрузить эталонное изображение!"));
    }
}

// Запуск потоку порівняння зображень
void CMFCDiplomDlg::StartComparison()
{
    m_running = true;
    m_thread = std::thread(&CMFCDiplomDlg::CompareImages, this);
}

// Зупинка потоку порівняння зображень
void CMFCDiplomDlg::StopComparison()
{
    m_running = false;
    if (m_thread.joinable())
```

```
    {
        m_thread.join();
    }
}

// Функція порівняння зображень
void CMFCDiplomDlg::CompareImages()
{
    for (int i = 0; m_running && i < 10; ++i)
    {
        CString filePath;
        filePath.Format(_T("E:\\MFC_Diplom\\converted_images\\image%d.jpg"), i);

        CStringA currentImagePath(filePath);
        cv::Mat currentImage = cv::imread(currentImagePath.GetString());

        if (currentImage.empty())
        {
            CString errorMessage;
            errorMessage.Format(_T("Не удалось загрузить изображение: %s"),
filePath);
            AfxMessageBox(errorMessage);
            continue;
        }

        CString result;
        detectMotion(m_refImage, currentImage, result);

        // Оновлення UI (виконується в основному потоці)
        CString finalResult = result;
        AfxGetMainWnd()->PostMessage(WM_USER_UPDATE_UI, 0,
reinterpret_cast<LPARAM>(new CString(finalResult)));

        LoadAndDisplayImage(filePath);

        std::this_thread::sleep_for(std::chrono::seconds(10));
    }
}

// Цей код для обробника події OnInitDialog, щоб завантажити еталонне зображення
під час запуску
BOOL CMFCDiplomDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // Додаткова ініціалізація
    LoadReferenceImage();  // Завантаження еталонного зображення
    LoadAndDisplayImage(CString(_T("E:\\MFC_Diplom\\reference.jpg")));
    TestImageLoading();

    return TRUE;
}

// Обробник кнопки "Захисні дані"
void CMFCDiplomDlg::OnBnClickedButtonSensor()
{
    if (m_running)
    {
        StopComparison();
        SetDlgItemText(IDC_BUTTON_SENSOR, _T("Запустити датчики"));
    }
    else
    {
        StartComparison();
        SetDlgItemText(IDC_BUTTON_SENSOR, _T("Зупинити датчики"));
    }
}

// Обробник повідомлення для оновлення UI
```

```
afx_msg LRESULT CMFCDiplomDlg::OnUpdateUI(WPARAM wParam, LPARAM lParam)
{
    CString* pResult = reinterpret_cast<CString*>(lParam);
    if (pResult)
    {
        m_infoControl.SetWindowText(*pResult);
        delete pResult;
    }
    return 0;
}

BEGIN_MESSAGE_MAP(CMFCDiplomDlg, CDialogEx)
    // Інші повідомлення
    ON_MESSAGE(WM_USER_UPDATE_UI, &CMFCDiplomDlg::OnUpdateUI)
END_MESSAGE_MAP()
```

integral.h – заголовочний файл

```
/*************************************************************
*  --- Програма системи ідентифікації на основі розпізнавання відбитку долоні користувача---                                          *
*
*                                                                 *
*************************************************************/

#ifndef INTEGRAL_H
#define INTEGRAL_H

#include <algorithm>  // запит для std::min/max

// невизначений VS макрос
#ifdef min
  #undef min
#endif

#ifdef max
  #undef max
#endif

#include <cv.h>

//! розраховуємо цілочисельне зображення  з image img.
IplImage *Integral(IplImage *img);


//! Обчислюємо суму пікселів в межах прямокутника, вказаного верхнім лівим, запускаємо координату і розмір
inline float BoxIntegral(IplImage *img, int row, int col, int rows, int cols)
{
  float *data = (float *) img->imageData;
  int step = img->widthStep/sizeof(float);

  // Віднімання для рядків/колонок.
  int r1 = std::min(row,          img->height) - 1;
  int c1 = std::min(col,          img->width)  - 1;
  int r2 = std::min(row + rows,   img->height) - 1;
  int c2 = std::min(col + cols,   img->width)  - 1;

  float A(0.0f), B(0.0f), C(0.0f), D(0.0f);
  if (r1 >= 0 && c1 >= 0) A = data[r1 * step + c1];
  if (r1 >= 0 && c2 >= 0) B = data[r1 * step + c2];
  if (r2 >= 0 && c1 >= 0) C = data[r2 * step + c1];
  if (r2 >= 0 && c2 >= 0) D = data[r2 * step + c2];

  return std::max(0.f, A - B - C + D);
}

#endif
```

Diplom_eye.cpp

```
#include "pch.h"
#include "framework.h"
#include "Diplom_eye.h"
#include "Diplom_eyeDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// CMFCDiplomApp

BEGIN_MESSAGE_MAP(CMFCDiplomApp, CWinApp)
    ON_COMMAND(ID_HELP, &CWinApp::OnHelp)
END_MESSAGE_MAP()

// Создание CMFCDiplomApp

CMFCDiplomApp::CMFCDiplomApp()
{
    // Підтримка диспетчера перезавантаження
    m_dwRestartManagerSupportFlags = AFX_RESTART_MANAGER_SUPPORT_RESTART;

    // TODO: додайте код створення,
    // Розміщує весь важливий код ініціалізації в InitInstance
}

// Єдиний об’єкт CMFCDiplomApp

CMFCDiplomApp theApp;

// Ініціалізація CMFCDiplomApp

BOOL CMFCDiplomApp::InitInstance()
{
    // InitCommonControlsEx() потрібно для Windows XP, якщо маніфест
    // Додаток використовує ComCtl32.dll версії 6 або пізнішої версії для
включення
    // стилів відображення.  В іншому випадку виникатиме збій при створенні
будь-якого вікна.
    INITCOMMONCONTROLSEX InitCtrls;
    InitCtrls.dwSize = sizeof(InitCtrls);
    // Виберіть цей параметр для включення всіх загальних класів керування, які
потрібно використовувати
    // у вашому додатку.
    InitCtrls.dwICC = ICC_WIN95_CLASSES;
    InitCommonControlsEx(&InitCtrls);

    CWinApp::InitInstance();

    AfxEnableControlContainer();

    // Створити диспетчер оболонки, якщо діалогове вікно містить
    // Подання дерева оболонки або будь-які його елементи управління.
    CShellManager* pShellManager = new CShellManager;

    // Активація візуального диспетчера "Класичний Windows" для включення
елементів керування MFC

CMFCVisualManager::SetDefaultManager(RUNTIME_CLASS(CMFCVisualManagerWindows));

    // Стандартна ініціалізація
    // Якщо ці можливості не використовуються і необхідно зменшити розмір
    // кінцевого файлу, що виконується, необхідно видалити з наступних
    // конкретні процедури ініціалізації, які не потрібні
    // Змініть розділ реєстру, де зберігаються параметри
    // TODO: слід змінити цей рядок на щось відповідне,
    // наприклад, на назву організації
```

```
    SetRegistryKey(_T("Локальні програми, створені за допомогою майстра
програм"));

    CMFCDiplomDlg dlg;
    m_pMainWnd = &dlg;
    INT_PTR nResponse = dlg.DoModal();
    if (nResponse == IDOK)
    {
        // TODO: Введіть код для обробки закриття діалогового вікна
        // За допомогою кнопки "OK"
    }
    else if (nResponse == IDCANCEL)
    {
        // TODO: Введіть код для обробки закриття діалогового вікна
        // за допомогою кнопки "Скасувати"
    }
    else if (nResponse == -1)
    {
        TRACE(traceAppMsg, 0, "Попередження. Не вдалося створити діалогове
вікно, тому роботу програми несподівано завершено.\n");
        TRACE(traceAppMsg, 0, "Попередження. У разі використання елементів
керування MFC для діалогового вікна неможливо #define
_AFX_NO_MFC_CONTROLS_IN_DIALOGS.\n");
    }

    // Видалити диспетчер оболонки, створений вище.
    if (pShellManager != nullptr)
    {
        delete pShellManager;
    }

#if !defined(_AFXDLL) && !defined(_AFX_NO_MFC_CONTROLS_IN_DIALOGS)
    ControlBarCleanUp();
#endif

    // Оскільки діалогове вікно закрито, поверніть значення FALSE, щоб вийти з
    // Програми замість запуску генератора повідомлень програми.
    return FALSE;
}
```

ProtocolDlg.cpp

```
#include "pch.h"
#include "ProtocolsDlg.h"
#include "afxdialogex.h"
#include "Resource.h"

IMPLEMENT_DYNAMIC(CProtocolsDlg, CDialogEx)

CProtocolsDlg::CProtocolsDlg(CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_PROTOCOLS_DIALOG, pParent)
{
}

void CProtocolsDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_PROTOCOLS_LIST, m_protocolsList);
}

BEGIN_MESSAGE_MAP(CProtocolsDlg, CDialogEx)
    ON_LBN_SELCHANGE(IDC_PROTOCOLS_LIST,
&CProtocolsDlg::OnLbnSelchangeProtocolsList)
END_MESSAGE_MAP()

void CProtocolsDlg::OnLbnSelchangeProtocolsList()
{
    int sel = m_protocolsList.GetCurSel();
    if (sel != LB_ERR)
    {
        CString protocol;
        m_protocolsList.GetText(sel, protocol);
        AfxMessageBox(_T("Ви вибрали протокол: ") + protocol);
    }
}
```

SettingsDlg.cpp

```
#include "pch.h"
#include "SettingsDlg.h"
#include "afxdialogex.h"
#include "Resource.h"

IMPLEMENT_DYNAMIC(CSettingsDlg, CDialogEx)

CSettingsDlg::CSettingsDlg(CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_SETTINGS_DIALOG, pParent)
{
}

void CSettingsDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CSettingsDlg, CDialogEx)
                                END_MESSAGE_MAP()
```

Diplom_eye.h

```
#pragma once

#ifndef __AFXWIN_H__
	#error "включить pch.h до включения этого файла в PCH"
#endif

#include "resource.h"		// основні символи
#include <opencv2/opencv.hpp>

class CMFCDiplomApp : public CWinApp
{
public:
	CMFCDiplomApp();

// Перевизначення
public:
	virtual BOOL InitInstance();

// Реалізація

	DECLARE_MESSAGE_MAP()
};

extern CMFCDiplomApp theApp;

void detectMotion(const cv::Mat& refImage, const cv::Mat& currentImage, CString&
result);
```

Diplom_eyeDlg.h

```
#pragma once

#include "afxwin.h"
#include "Resource.h"
#include <gdiplus.h>  // Включение заголовочного файла GDI+
#pragma comment (lib, "Gdiplus.lib")

class CMFCDiplomDlg : public CDialogEx
{
public:
    CMFCDiplomDlg(CWnd* pParent = nullptr);
    virtual ~CMFCDiplomDlg();  // Объявление деструктора

#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_MFC_DIPLOM_DIALOG };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);

protected:
    HICON m_hIcon;
    CStatic m_pictureControl;
    CStatic m_infoControl;

    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    DECLARE_MESSAGE_MAP()

public:
    afx_msg void OnBnClickedButtonSensor();
    afx_msg void OnBnClickedButtonSettings();
    afx_msg void OnBnClickedButtonDeveloper();
    afx_msg void OnBnClickedButtonProtocol();
    afx_msg void LoadAndDisplayImage(CString filePath);  // Оголошення функції
    afx_msg void TestImageLoading();  // Оголошення функції
    afx_msg LRESULT OnUpdateUI(WPARAM wParam, LPARAM lParam);  // Обробник
повідомлень для оновлення UI

private:
    Gdiplus::GdiplusStartupInput gdiplusStartupInput;
    ULONG_PTR gdiplusToken;

    cv::Mat m_refImage;  // Еталонне зображення
    std::atomic<bool> m_running;  // Прапорець для керування потоком
    std::thread m_thread;  // Потік для порівняння зображень

    void LoadReferenceImage();  // Завантаження еталонного зображення
    void StartComparison();  // Запуск потоку порівняння зображень
    void StopComparison();  // Зупинка потоку порівняння зображень
    void CompareImages();  // Функція порівняння зображень
};
```

framework.h

```
#pragma once

#ifndef VC_EXTRALEAN
#define VC_EXTRALEAN            // Виключіть рідко використовувані компоненти з
заголовків Windows
#endif

#include "targetver.h"

#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS      // деякі конструктори CString
будуть явними

// вимикає функцію приховування деяких загальних і часто пропусканих попереджень
MFC
#define _AFX_ALL_WARNINGS

#include <afxwin.h>         // основні та стандартні компоненти MFC
#include <afxext.h>         // Розширення MFC

#include <afxdisp.h>        // класи автоматизації MFC

#ifndef _AFX_NO_OLE_SUPPORT
#include <afxdtctl.h>           // підтримка MFC для типових елементів
управління Internet Explorer 4
#endif
#ifndef _AFX_NO_AFXCMN_SUPPORT
#include <afxcmn.h>             // підтримка MFC для типових елементів
управління Windows
#endif // _AFX_NO_AFXCMN_SUPPORT

#include <afxcontrolbars.h>     // підтримка MFC для стрічок і панелей
управління
```

ProtocolsDlg.h

```
#pragma once

class CProtocolsDlg : public CDialogEx
{
public:
    CProtocolsDlg(CWnd* pParent = nullptr);

#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_PROTOCOLS_DIALOG };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);

protected:
    DECLARE_MESSAGE_MAP()
public:
    CListBox m_protocolsList;
    afx_msg void OnLbnSelchangeProtocolsList();
};
```

Resource.h

```
//{{NO_DEPENDENCIES}}
// Включаєний файл, створений у Microsoft Visual C++.
// Використовується MFCDiplom.rc

#define IDM_ABOUTBOX                    0x0010
#define IDD_ABOUTBOX                    100
#define IDS_ABOUTBOX                    101
#define IDD_MFC_DIPLOM_DIALOG           102
#define IDR_MAINFRAME                   128
#define IDC_STATIC_PICTURE              1000
#define IDC_STATIC_PICTURE1             1001
#define IDC_BUTTON_SENSOR               1002
#define IDC_BUTTON_SETTINGS             1003
#define IDC_BUTTON_DEVELOPER            1004
#define IDC_STATIC_INFO                 1005
#define IDC_BUTTON_DEVELOPER2           1006
#define IDC_BUTTON_PROTOCOL             1006
#define IDD_SETTINGS_DIALOG             130
#define IDD_PROTOCOLS_DIALOG             140
#define IDC_PROTOCOLS_LIST               1007
#define WM_USER_UPDATE_UI (WM_USER + 1)



// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        131
#define _APS_NEXT_COMMAND_VALUE         32771
#define _APS_NEXT_CONTROL_VALUE         1006
#define _APS_NEXT_SYMED_VALUE           101
#endif
#endif
```

SettingsDlg.h

```
#pragma once

class CSettingsDlg : public CDialogEx
{
public:
    CSettingsDlg(CWnd* pParent = nullptr);

#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_SETTINGS_DIALOG };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);

protected:
    DECLARE_MESSAGE_MAP()
};
```