

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

СТАНІСЛАВЧУК ДЕНИС ОЛЕКСАНДРОВИЧ

Допускається до захисту:

Завідувач кафедри прикладної
математики та кібербезпеки, д-р
філософії з математики

_____ Луценко А. В.

«__» _____ 20__ р.

РОЗРОБЛЕННЯ БАГАТОПОРТОВОГО КАНАЛУ ДЛЯ УПРАВЛІННЯ
БЕЗПІЛОТНИМИ ЛІТАЛЬНИМИ АПАРАТАМИ

Спеціальність 125 Кібербезпека

Кваліфікаційна (бакалаврська) робота

Керівник:

Прокоф'єв М. І.,
професор кафедри прикладної
математики та кібербезпеки

(підпис)

Оцінка : _____ / _____ / _____

(бали/за шкало ЄКТС/за національною шкалою)

Голова ЕК: _____ (підпис)

Вінниця – 2025

ЗМІСТ

АНОТАЦІЯ	3
ВСТУП.....	4
РОЗДІЛ I. ТЕОРЕТИЧНІ ВІДОМОСТІ.....	6
1.1 FPV-дрони	6
1.2 Програмне та апаратне забезпечення	8
1.3 Концепція багатопортового каналу зв'язку для керування БПЛА ...	13
1.4 Висновки до першого розділу	16
РОЗДІЛ II. МЕТОДИКА СТВОРЕННЯ ЗАХИЩЕНОГО КАНАЛУ	18
2.1 Аналіз загроз та вимог безпеки.....	18
2.2 Проектування багатопортового каналу зв'язку	19
2.3 Розробка програмного забезпечення	22
2.4 Інтеграція та тестування.....	26
2.5 Оцінка результатів	29
2.6 Висновки до другого розділу	30
РОЗДІЛ III. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАХИЩЕНОГО КАНАЛУ	32
3.1 Код для управління ретрансляторами наземної станції.....	32
3.2 Код для управління ретрансляторами, що безпосередньо передають сигнали управління БПЛА	36
3.3 Код для точки ретрансляції	40
3.4 Висновки до третього розділу	42
ВИСНОВКИ	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	45

АНОТАЦІЯ

Станіславчук Д. О. Розроблення багатопортового каналу для управління безпілотними літальними апаратами. Спеціальність 125 Кібербезпека . Донецький національний університет імені Василя Стуса. Вінниця. 2025 рік.

У кваліфікаційній роботі було досліджено можливість реалізації багатопортового каналу зв'язку для управління безпілотними літальними апаратами через точки ретрансляції. Проведено аналіз технічних характеристик та існуючого програмного забезпечення для управління БПЛА.

Ключові слова: БПЛА, багатопортовий канал зв'язку, радіoeлектронна боротьба, шифрування, динамічна зміна точок трансляції.

Структура роботи: анотація, вступ, три розділи, висновки, список використаних джерел.

48 сторінок, 8 рис., 24 джерела

ABSTRACT

Stanislavchuk D. O. Development of a multi-port channel for controlling unmanned aerial vehicles. Specialty 125 Cybersecurity. Vasyl Stus Donetsk National University. Vinnytsia. 2025.

The thesis explores the possibility of implementing a multi-port communication channel for controlling unmanned aerial vehicles via relay points. An analysis of the technical characteristics and existing software for controlling UAVs was conducted.

Keywords: UAV, multi-port communication channel, electronic warfare, encryption, dynamic change of relay points.

Structure of the work: abstract, introduction, three chapters, conclusions, list of references.

48 pages, 8 figures, 24 references

ВСТУП

Актуальність роботи. У сучасних умовах безпілотні авіаційні системи (БПЛА) набули статусу незамінних інструментів у широкому спектрі застосувань, включаючи військову справу (ведення розвідки, здійснення спостереження, нанесення високоточних ударів) та забезпечення безпеки (патрулювання територій, проведення пошуково-рятувальних операцій). Ця екстенсивна інтеграція БПЛА у критично важливі сфери діяльності зумовлює їхню високу цінність як стратегічних активів, що, своєю чергою, робить їх привабливими цілями для потенційних противників, які прагнуть дестабілізувати їхнє функціонування або використовувати у власних деструктивних цілях.

Традиційні системи управління БПЛА, що базуються на одноканальному зв'язку з фіксованими географічними координатами точок трансляції сигналу, демонструють значну вразливість до виявлення та придушення засобами радіоелектронної боротьби (РЕБ) противника. Компрометація каналу управління наражає БПЛА на критичні ризики, включаючи втрату оперативного контролю, зрив виконання поставлених завдань та потенційну можливість несанкціонованого захоплення апарату. В умовах складного оперативного середовища або активної протидії противника, забезпечення надійного та безпечного зв'язку з БПЛА набуває першочергового значення.

Одним із перспективних підходів до підвищення живучості систем управління БПЛА є маскуваність пунктів управління шляхом використання динамічних точок трансляції сигналу. Оперативна зміна географічного розташування ретрансляторів сигналу ускладнює противнику процес пеленгації та цілеспрямованого радіоелектронного впливу на канал управління. Водночас, стрімкий розвиток технологій радіоелектронної боротьби вимагає розробки новітніх методів захисту, що перевищують ефективність традиційних контрзаходів. У цьому контексті, розробка стійких до РЕБ систем управління, що використовують, зокрема, багатопортові канали зв'язку, є критично важливим напрямком для забезпечення безперервного та ефективного управління БПЛА в умовах активної протидії.

Захищений канал зв'язку, окрім маскуванню пункту управління, також забезпечує конфіденційність переданих даних, запобігаючи несанкціонованому доступу, перехопленню та маніпулюванню критично важливою інформацією. Враховуючи зростання спектру та інтенсивності потенційних загроз, забезпечення надійного, безпечного та ефективного використання БПЛА є нагальною науково-технічною проблемою.

Мета роботи. Метою даного дослідження є розробка концепції та принципів функціонування захищеного багатопортового каналу управління для маскуванню наземних станцій управління БПЛА шляхом реалізації динамічної зміни точок трансляції сигналу.

Технічне завдання. Для досягнення поставленої мети передбачається розробка програмного забезпечення, що забезпечуватиме динамічну зміну точок трансляції сигналу для управління безпілотними авіаційними системами. Для цього слід розробити алгоритм вибору та переключення каналів управління між доступними точками трансляції, враховуючи критерії безпеки, надійності зв'язку та протидії РЕБ.

РОЗДІЛ I. ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 FPV-дрони

FPV-дрон - клас безпілотних авіаційних систем, управління якими здійснюється оператором через інтегровану систему візуалізації, що забезпечує перспективу від першої особи (First Person View) з бортової камери літального апарату. Передача відеосигналу в режимі реального часу на спеціалізовані окуляри (FPV-шолом) або дисплей створює для оператора імерсивний ефект присутності в кабіні пілота, що значно розширює можливості керування та застосування даної технології [1].

Канали та частоти, що використовуються в FPV-дронах:

Найчастіше, для управління використовуються частотні діапазони 2.4 ГГц та 900 МГц. Діапазон 2.4 ГГц є більш поширеним, але може бути більш схильним до перешкод, оскільки використовується багатьма іншими бездротовими пристроями (Wi-Fi, Bluetooth тощо). Діапазон 900 МГц має кращу проникність та дальність, але може мати законодавчі обмеження в деяких регіонах. Також набирають популярності системи керування на базі протоколу ELRS (ExpressLRS), які можуть працювати на частотах 2.4 ГГц або 915/868 МГц, забезпечуючи високу надійність та низьку затримку [2].

Найпоширеніші частотні діапазони для передачі відео:

- **5.8 ГГц:** Забезпечує високу якість зображення та низьку затримку, але має меншу дальність та більш чутливий до перешкод (будівлі, дерева). Є найбільш популярним для гонок та фрістайлу.

- **1.2 ГГц / 1.3 ГГц:** Мають більшу дальність та кращу проникність порівняно з 5.8 ГГц, але можуть мати нижчу якість відео та законодавчі обмеження.

- **900 МГц:** Ще більша дальність та проникність, але якість відео може бути ще нижчою, і використання може бути обмежено законодавством.

- Цифрові системи (DJI O3, Walksnail Avatar, HDZero): Використовують власні протоколи та частотні діапазони (зазвичай в діапазонах 5.8 ГГц), забезпечуючи високу якість HD-відео, низьку затримку та стійкість до перешкод [3].

Основні моделі FPV, які використовуються в ЗСУ

DJI Mavic 3E (Enterprise) являє собою передовий комерційний дрон, розроблений для виконання широкого спектра завдань, включаючи картографування, інспекцію та розвідку. Його ключовими особливостями є висока продуктивність камери, розширені функції безпеки та ефективна система передачі даних [4].

Таблиця 1. Характеристики DJI Mavic 3E (Enterprise)

Характеристика	Значення
Маса	0,915 кг
Максимальна злітна маса	1,05 кг
Розміри	У складеному стані: 221×96,3×90,3 мм (Д×Ш×В). У розкладеному стані (без пропелерів): 347,5×283×107,7 мм (Д×Ш×В).
Максимальна швидкість польоту	15 м/с (звичайний режим)
Максимальна стійкість до вітру	12 м/с
Максимальна висота польоту над рівнем моря	6000 м
Максимальний час польоту	До 45 хвилин
Максимальний час зависання	До 38 хвилин
Максимальна дальність польоту	До 32 км
Робочі частоти	2.400-2.4835 ГГц, 5.725-5.850 ГГц

DJI Mavic 3T (Thermal) виділяється наявністю тепловізійної камери на додаток до звичайної та телекамери [5].

Таблиця 2. Характеристики DJI Mavic 3T

Характеристика	Значення
Маса	0,92 кг
Максимальна злітна маса	1,05 кг
Розміри	У складеному стані: 221×96,3×90,3 мм (Д×Ш×В). У розкладеному стані (без пропелерів): 347,5×283×107,7 мм (Д×Ш×В).

Максимальна швидкість польоту	15 м/с (звичайний режим)
Максимальна стійкість до вітру	12 м/с
Максимальна висота польоту над рівнем моря	6000 м
Максимальний час польоту	До 45 хвилин
Максимальний час зависання	До 38 хвилин
Максимальна дальність польоту	До 32 км
Робочі частоти	2.400-2.4835 ГГц, 5.725-5.850 ГГц

DJI Mavic 2 Enterprise Advanced - універсальний комерційний дрон, відомий своїми передовими подвійними тепловізійними і візуальними камерами та високоточним позиціонуванням [6].

Таблиця 3. Характеристики DJI Mavic 2 Enterprise Advanced

Характеристика	Значення
Маса	0,909 кг
Максимальна злітна маса	1,1 кг
Розміри	У складеному стані: 214×91×84 мм (Д×Ш×В). У розкладеному стані (без пропелерів): 322×242×84 мм (Д×Ш×В).
Максимальна швидкість польоту	14 м/с (звичайний режим)
Максимальна стійкість до вітру	12 м/с
Максимальна висота польоту над рівнем моря	6000 м
Максимальний час польоту	До 31 хвилин
Максимальний час зависання	До 30 хвилин
Максимальна дальність польоту	До 24 км
Робочі частоти	2.400-2.4835 ГГц, 5.725-5.850 ГГц

1.2 Програмне та апаратне забезпечення

PX4 Autopilot є open-source автопіотною системою з модульною архітектурою, розробленою для автономних безпілотних літальних апаратів (БПЛА).

Основні компоненти

Flight Stack - це ядро автопіотної системи PX4, що реалізує алгоритми керування польотом БПЛА та забезпечує всі необхідні функції для стабілізації,

навігації, керування та виконання місії літальним апаратом. Його можна розглядати як вертикальну ієрархію програмних компонентів, кожен з яких відповідає за певний набір завдань і взаємодіє з сусідніми рівнями.

Суть Flight Stack полягає в абстракції складності та модульності, що полегшує розробку, тестування, налагодження та розширення функціональності системи. Кожен рівень має чітко визначені вхідні та вихідні дані, а також функціональні обов'язки [7].

Критично важливим елементом, що забезпечує детерміноване виконання завдань реального часу, є використання операційної системи реального часу (RTOS - Real-Time Operating System) як базової платформи для побудови Flight Stack. RTOS гарантує передбачуваність виконання критично важливих процесів з мінімальними часовими затримками, що є імперативом для забезпечення безпеки та ефективності керування польотом.

Flight Stack містить наступні ключові функціональні модулі для реалізації місії:

- **State Estimator (Модуль оцінки стану):** даний модуль відповідає за інтеграцію даних, отриманих від різномірних бортових сенсорів (інерціальних вимірювальних блоків, глобальних навігаційних супутникових систем, барометрів тощо), використовуючи складні алгоритми фільтрації (наприклад, фільтр Калмана та його модифікації). Його метою є формування найбільш точної та надійної оцінки поточного динамічного стану літального апарату, включаючи його просторове положення, орієнтацію, лінійні та кутові швидкості, а також їхню часову екстраполяцію [7, 8].

- **Flight Controller (Модуль керування польотом):** цей модуль отримує інформацію про заданий бажаний стан (визначений оператором або системою автономної навігації) та поточну оцінку стану від модуля State Estimator. На основі цих вхідних даних, використовуючи алгоритми керування (наприклад, PID-регулятори, модельно-орієнтоване керування), він генерує керуючі сигнали для виконавчих механізмів (електродвигунів, сервоприводів), спрямовані на

мінімізацію розбіжностей між цільовим та фактичним станом, забезпечуючи стабільність та керованість апарату в процесі польоту [9].

- **Navigation System (Система автономної навігації):** даний модуль відповідає за планування оптимальних траєкторій руху, визначення навігаційних цілей та здійснення автономного керування переміщенням літального апарату. Він використовує інформацію про задані місії, дані глобальних навігаційних супутникових систем (за наявності), результати оцінки стану та навігаційні алгоритми для генерації послідовності проміжних точок маршруту та команд керування для модуля Flight Controller. [7]

Middleware – це шар програмного забезпечення, який розташовується між операційною системою та прикладними програмами. Воно виконує роль посередника, забезпечуючи зв'язок, керування даними, автентифікацію, обробку запитів та інші сервісні функції, які є загальними для різних застосунків [10].

У PX4 до Middleware відносять uORB (micro Object Request Broker) – система міжпроцесної взаємодії та MAVLink

PX4 підтримує широкий спектр апаратних засобів для керування польотом, включаючи різні типи льотних контролерів (autopilots), сенсори (IMU, GPS, компаси, барометри), системи телеметрії та інші периферійні пристрої [11].

Micro Air Vehicle Link (MAVLink) - це легкий протокол прикладного рівня передачі даних, розроблений для комунікації між безпілотними авіаційними системами та наземними контрольними станціями.

MAVLink працює за сучасною гібридною схемою "публікація-підписка" та "точка-точка": Потоки даних надсилаються/публікуються як теми, в той час як підпротоколи конфігурації, такі як протокол місії або протокол параметрів, передаються "точка-точка" з ретрансляцією [12].

Таблиця 4. Структура пакету протоколу

Назва поля	Індекс позиції	Призначення
Start-of-frame	0	Помічає початок передачі фрейму
Payload-length	1	Довжина корисного навантаження

Packet sequence	2	Кожен компонент підраховує порядковий номер відправлених пакетів у послідовності. Дозволяє відслідковувати втрату пакету
System ID	3	Ідентифікатор системи, що відсилає дані. Дозволяє відрізнити різні системи в одній мережі.
Component ID	4	Ідентифікатор компонента, що відсилає дані. Дозволяє відрізнити різні компоненти однієї системи, наприклад, інерційний вимірювальний пристрій (IMU) та автопілот.
Message ID	5	Ідентифікатор типу повідомлення – повідомляє як дані в полі payload треба розуміти і як їх правильно декодувати.
Payload	6 до n+6	Дані повідомлення, залежать від ідентифікатору повідомлення – message id.
CRC	n+7 до n+8	Контрольна сума всього пакету, включаючи стартовий символ (LSB, MSB)

UART (Universal Asynchronous Receiver/Transmitter) – це асинхронний апаратний протокол послідовної комунікації, який використовується для обміну даними між двома пристроями. Він є одним із найпростіших і найдавніших протоколів послідовної передачі даних і досі широко застосовується в різних вбудованих системах, мікроконтролерах та комп'ютерах для зв'язку з периферійними пристроями.

У безпілотниках, UART часто використовується як фізичний рівень передачі даних між автопілотом (наприклад, на базі PX4 або ArduPilot) та наземною станцією керування або іншими бортовими пристроями. MAVLink визначає формат повідомлень, які передаються через цей фізичний канал. UART підходить для обміну даними з сенсорами, GPS-модулями, Bluetooth-модулями, радіоприймачами малої потужності та іншими пристроями, які не вимагають високої пропускної здатності каналу [13].

Таблиця 5. Формат кадру UART

Назва поля	Індекс позиції	Призначення
Start Bit	0	сигналізує початок передачі байта даних
Data Bits	1-5 або 1-9	представляють передаваний символ або частину даних

Parity Bit	10	використовується для простого виявлення помилок передачі
Stop Bits	11-12	сигналізують кінець передачі кадру та забезпечують інтервал між послідовними кадрами

User Datagram Protocol (UDP) являє собою простий протокол транспортного рівня, що функціонує поверх протоколу Інтернет (IP) у стеку TCP/IP. Його ключовою характеристикою є відсутність встановлення з'єднання (connectionless) та ненадійність (unreliable) в обміні дейтаграмами [14].

Таблиця 6. Формат кадру UDP

Назва поля	Індекс позиції	Призначення
Source Port	0-15	Це поле ідентифікує порт відправника
Destination Port	16-31	ідентифікує порт одержувача
Length	32-47	вказує довжину в байтах UDP-дейтаграми
Checksum	48-63	може використовуватися для перевірки помилок у заголовку та даних

UDP часто використовується для передачі поточкових телеметричних даних (положення, орієнтація, стан сенсорів) від БАС до НСК через бездротові IP-мережі (наприклад, Wi-Fi, стільникові мережі). Також для передачі відеопотоку з бортової камери на НСК або інші приймачі. UDP забезпечує низьку затримку, що є важливим для керування FPV-дронами [15].

SPI (Serial Peripheral Interface) – синхронний послідовний протокол комунікації, що функціонує за принципом головний-підлеглий (Master-Slave) та призначений для обміну даними на обмежених відстанях між інтегральними схемами, зокрема мікроконтролерами та периферійними пристроями [16].

У дронах SPI є критично важливим інтерфейсом для зв'язку між центральним контролером польоту та різноманітними бортовими периферійними пристроями, включаючи:

- **Інерціальні вимірювальні блоки (IMU):** Для отримання високочастотних даних про кутові швидкості та лінійні прискорення, необхідних для стабілізації та керування орієнтацією апарату.

- **Барометричні датчики:** Для прецизійного вимірювання атмосферного тиску з метою визначення висоти польоту.

- **Магнетометри:** Для визначення абсолютного курсу апарату шляхом вимірювання магнітного поля Землі.

- **Приймачі супутникових навігаційних систем (GNSS):** У деяких випадках для високошвидкісної передачі сирих навігаційних даних.

- **Пристрої відображення телеметричної інформації на екрані (OSD):** Для інтеграції бортових даних у відеопотік.

- **Інтерфейси карт пам'яті:** Для запису льотних даних, параметрів конфігурації та іншої інформації [17].

RTPS (Real-Time Publish Subscribe Protocol) — це протокол передачі даних в реальному часі, що базується на моделі "видавець-підписник". Він розроблений для забезпечення надійної та ефективної комунікації в системах, які вимагають обміну даними в реальному часі, навіть через ненадійні транспортні мережі, такі як UDP [18].

1.3 Концепція багатопортового каналу зв'язку для керування БПЛА

Концепція багатопортового каналу зв'язку для керування БПЛА передбачає використання кількох радіочастотних каналів (портів) одночасно або з можливістю швидкого переключення між ними для передачі команд управління та отримання телеметричних даних від безпілотного літального апарату до наземної станції управління і навпаки.

Багатопортовий канал зв'язку для керування БПЛА надає значні переваги, насамперед у підвищенні надійності, стійкості та безпеки системи керування в умовах складного радіоелектронного середовища. Підвищена надійність досягається завдяки резервуванню каналів, де наявність кількох незалежних каналів забезпечує безперервну передачу критично важливих команд управління та телеметрії навіть у випадку тимчасових проблем на одному з каналів, значно

знижуючи ризик повної втрати зв'язку. Ймовірність одночасної втрати зв'язку на кількох каналах є значно нижчою, що забезпечує стабільніше керування [19].

Підвищена стійкість проявляється у здатності протистояти радіоелектронним завадам, оскільки система може автоматично переключатися на вільні від перешкод частотні діапазони, ускладнюючи противнику блокування каналу управління. Багатопортова система також адаптивна до умов радіоефіру, динамічно обираючи оптимальні частоти.

Підвищена безпека забезпечується ускладненням перехоплення сигналу через розподіл інформації між кількома каналами на різних частотах та протоколах, що вимагає від зловмисника одночасного моніторингу та дешифрування кількох каналів. Захист від підміни команд також зростає, оскільки зловмиснику необхідно імітувати сигнали на кількох частотах одночасно. Крім того, кожен порт може використовувати різні методи шифрування, підвищуючи загальну стійкість до криптоаналізу [20].

Додатково, багатопортовість може збільшити пропускну здатність для передачі великих обсягів даних та покращити якість обслуговування, дозволяючи передавати різні типи даних з різними пріоритетами через різні порти. В умовах сучасної війни, особливо в Україні, де активно застосовується РЕБ, впровадження концепції багатопортового каналу зв'язку є критично важливим для забезпечення успішного виконання місій, збереження БПЛА та безпеки операторів.

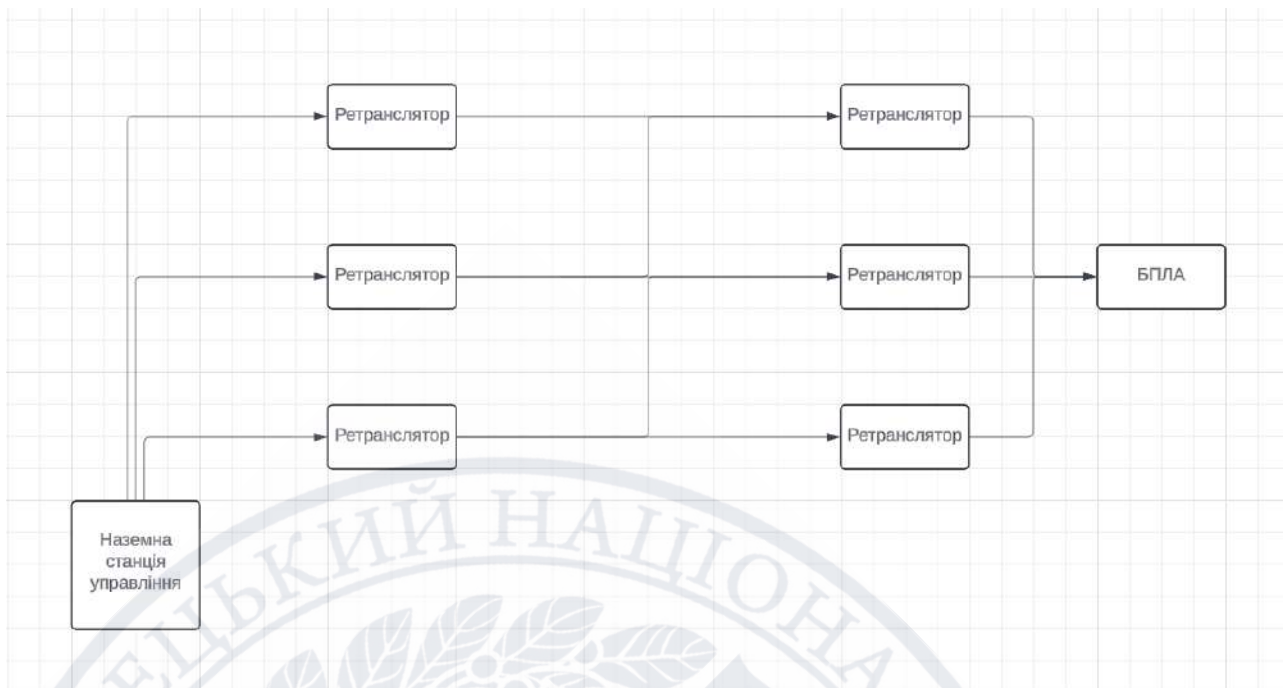


Рисунок 1.1 Блок-схема багатопортового каналу

Декілька ретрансляторів на певній відстані розміщують у просторі один від одного (рис.1.1). Сигнал йде від наземною станції управління через три можливих канали певним шляхом, який обирається програмно. Він визначається за допомогою ПЗ, яке визначає та присвоює певні частоти для каналів (портів).

Основні пункти реалізації:

- Захист наземної станції управління через фізичне відокремлення передавальної антени від пункту управління за допомогою дротового (мідного), волоконно-оптичного кабелю, або оптичного каналів. Це унеможливує пряме радіоелектронне виявлення місцезнаходження операторів, що здійснюють оперативне управління безпілотним апаратом.

- Передача сигналу на відстань з використанням волоконно-оптичного кабелю, або оптичного каналу забезпечує високу швидкість передачі даних, низький рівень електромагнітних завад та значну дальність зв'язку, особливо при використанні потужних медіаконверторів та оптичних підсилювачів.

- Безпілотний літальний апарат виступає в ролі ретранслятора, приймаючи захищений сигнал від віддаленої антени та передаючи його на інші БПЛА.

- Динамічна зміна точки ретрансляції за допомогою хаба (комутатора) для підключення кількох медіаконверторів, кожен з яких може бути підключений до

окремої віддаленої антени, дозволяє оперативно змінювати фізичну точку випромінювання радіосигналу. Це значно ускладнює пеленгацію джерела сигналу противником та підвищує живучість системи управління.

1.4 Висновки до першого розділу

В даному розділі було розглянуто теоретичні аспекти, необхідні для розуміння принципів побудови захищеного багатопортового каналу управління БПЛА.

Насамперед, було розглянуто клас FPV-дронів, їхнє призначення та характерні особливості. FPV-дрони, управління якими здійснюється з використанням окулярів віртуальної реальності, забезпечують оператору унікальний досвід "польоту від першої особи". Це відкриває нові можливості для застосування БПЛА, але також ставить підвищені вимоги до надійності та безпеки каналу зв'язку. У зв'язку з цим, було досліджено архітектуру FPV-дронів, а також частотні діапазони, що використовуються для управління та передачі відеосигналу (2.4 ГГц, 900 МГц, 5.8 ГГц, 1.2/1.3 ГГц). Було відзначено, що вибір частотного діапазону є компромісом між дальністю зв'язку, пропускнуою здатністю, стійкістю до перешкод та законодавчими обмеженнями.

Далі, було проаналізовано програмне та апаратне забезпечення, яке є невід'ємною частиною сучасних БПЛА. Розглянуто open-source автопілотну систему PX4, її модульну архітектуру та ключові компоненти, такі як Flight Stack, Middleware (uORB, MAVLink) та операційна система реального часу (RTOS). Окрему увагу було приділено протоколам передачі даних, що використовуються для зв'язку між БПЛА та наземною станцією управління, а саме: MAVLink, UART, UDP, SPI та RTPS. Розуміння особливостей цих протоколів є важливим для розробки ефективної та надійної системи зв'язку.

Було розглянуто основну ідею багатопортовості, яка полягає у використанні кількох незалежних радіочастотних каналів для підвищення надійності, стійкості та безпеки системи управління БПЛА. Було проаналізовано

переваги багатопортового підходу, такі як резервування каналів, стійкість до радіоелектронних завад, ускладнення перехоплення сигналу та можливість збільшення пропускної здатності. Також було окреслено основні принципи реалізації багатопортового каналу, включаючи захист наземної станції управління, передачу сигналу на відстань, використання БПЛА як ретранслятора та динамічну зміну точки ретрансляції.

За матеріалами, що проаналізовані у першому розділі, обґрунтовані напрямки для подальшої розробки та проектування захищеного багатопортового каналу управління БПЛА.



РОЗДІЛ II. МЕТОДИКА СТВОРЕННЯ ЗАХИЩЕНОГО КАНАЛУ

Створення захищеного багатопортового каналу управління БПЛА з динамічною зміною точок трансляції сигналу включає наступні етапи:

- Етап 1: Аналіз загроз та вимог безпеки
- Етап 2: Проектування багатопортового каналу зв'язку
- Етап 3: Розробка програмного забезпечення
- Етап 4: Інтеграція та тестування
- Етап 5: Оцінка результатів

2.1 Аналіз загроз та вимог безпеки

На етапі ідентифікації потенційних загроз проводиться дослідження можливих негативних впливів на канал управління БПЛА, які можуть призвести до втрати контролю, зриву місії, витоку інформації або несанкціонованого використання апарату.

Основні загрози:

- Радіоелектронне придушення каналу управління
- Пеленгація та визначення місцезнаходження наземної станції управління
- Перехоплення та дешифрування команд управління та телеметричних даних

- Атаки типу Man-in-the-Middle

- Фізичне захоплення компонентів системи

Вимоги безпеки:

- Стійкість каналу управління до радіоелектронних завад
- Маскування місцезнаходження наземної станції управління
- Конфіденційність переданих даних
- Автентифікація та авторизація учасників зв'язку
- Цілісність переданих даних
- Оперативність переключення між точками трансляції

2.2 Проєктування багатопортового каналу зв'язку

Проєктування багатопортового каналу зв'язку починається з вибору фізичного рівня зв'язку, що є важливим для забезпечення стійкості, пропускну здатності та безпеки каналу зв'язку.

Вибір радіочастотних діапазонів залежить від різних параметрів, найрозповсюдженішими та найдоцільнішими для FPV-дронів є частоти 2,4 ГГц та 5,8 ГГц які є неліцензованими та доступними для загального використання. Вибір конкретного діапазону залежить від вибору між доступністю, дальністю зв'язку, пропускну здатністю та стійкістю до завад.

Також потрібно обрати модуляційну схему. До основних належать: OFDM, FHSS, DSSS

OFDM - це метод цифрової багатоканальної модуляції. Замість передачі одного високошвидкісного потоку даних на одній несучій частоті, OFDM ділить доступну смугу пропускання на безліч вузьких піднесучих, які є ортогональними одна до одної. Кожна піднесуча передає частину даних з нижчою швидкістю [22]. Основні переваги:

- Стійкість до багатопроменевого згасання
- Стійкість до вузькосмугових завад
- Ефективність використання спектра
- Адаптація до змінних умов каналу

FHSS — це метод розширення спектра, в якому несуча частота переданого сигналу швидко змінюється (стрибає) відповідно до псевдовипадкової послідовності, відомої як передавачу, так і приймачу. Сигнал займає вузьку смугу пропускання в будь-який момент часу, але ця центральна частота стрибає по ширшій смузі частот протягом часу [23]. Основні переваги:

- Стійкість до вузькосмугового глушіння
- Складність перехоплення
- Співіснування з іншими системами

- Низька ймовірність виявлення

DSSS — це ще один метод розширення спектра, в якому вихідний сигнал даних множить на високошвидкісний псевдовипадковий шум (PN-код), також відомий як чип-код. Цей процес розподіляє енергію сигналу по значно ширшій смузі пропускання, ніж вихідний сигнал даних [24]. Основні переваги:

- Стійкість до вузькосмугових завад і глушіння
- Підвищена безпека через PN-код
- Можливість множинного доступу з кодовим розділенням
- Точне визначення дальності та часу

Наступним кроком повинен бути вибір типу антен, це залежить від сценарію використання, необхідної дальності зв'язку та вимог до маскування.

Є два типи антен – спрямовані та всеспрямовані. Спрямовані антени дозволяють концентрувати енергію сигналу в певному напрямку, збільшуючи дальність зв'язку та зменшуючи ймовірність перехоплення та пеленгації з інших напрямків. Їх краще використовувати на наземних станціях управління та точках трансляції. Всеспрямовані антени забезпечують зв'язок незалежно від орієнтації БПЛА, але можуть бути більш вразливими до завад та мати меншу дальність. Їх краще використати на БПЛА та для забезпечення покриття точками трансляції.

Далі потрібно розробити архітектуру багатопортовості. Почнемо з визначення кількості портів - визначається компромісом між досяжним рівнем резервування, складністю реалізації та доступним спектром частот. Можливе використання 2-4 одночасних портів.

Вибір та розробка алгоритмів розподілу команд управління та телеметрії між портами:

- **Дублювання:** Одночасна передача одних і тих самих даних по кількох портах для підвищення ймовірності успішної доставки.

- **Розподіл за типом даних:** Передача критично важливих команд управління по кількох захищених портах, а менш критичної телеметрії – по одному або кількох інших портах.

- **Чергування:** Використання різних портів для послідовної передачі пакетів даних.

Наступний крок - проектування системи динамічної зміни точок трансляції. Потенційні географічні точки трансляції залежать від оперативного сценарію та доступних ресурсів. Для наземних станцій управління будуть використані стаціонарні точки - заздалегідь визначені та підготовлені місця розташування ретрансляторів. Для ретрансляції сигналу будуть використані мобільні точки - дрони, оснащені необхідним обладнанням для ретрансляції сигналу.

Критерії вибору оптимальної точки трансляції:

- **Рівень сигналу:** Вибір точки, що забезпечує найсильніший та найстабільніший сигнал до БПЛА.

- **Рівень завад:** Уникнення точок, розташованих в зонах високого рівня радіоелектронних завад.

- **Прогнозована загроза РЕБ:** Аналіз інформації про активність засобів РЕБ противника та вибір найменш вразливої точки.

- **Доступність каналу:** Перевірка працездатності каналу зв'язку через потенційну точку трансляції.

- **Географічне розташування:** Вибір точки, що забезпечує оптимальне покриття зони польоту БПЛА та ускладнює пеленгацію НСУ.

Наступним важливим кроком є розробка протоколів ініціації та виконання переключення між точками трансляції. В першу чергу протокол повинен забезпечувати надійну та швидку зміну активної точки без втрати контролю над БПЛА. Можна розглянути 2 основних способи: автоматичне переключення – може бути реалізовано на основі певних критеріїв (наприклад, погіршення рівня сигналу, виявлення завад, через певні інтервали часу) система автоматично ініціює переключення на іншу доступну точку, або переключення за командою оператора - оператор має можливість вручну ініціювати переключення на іншу точку трансляції.

Реалізація механізмів синхронізації переключення між НСУ та БПЛА повинні бути синхронізовані щодо поточної активної точки трансляції. Це може бути досягнуто шляхом обміну службовими повідомленнями або використання заздалегідь визначених послідовностей переключення.

2.3 Розробка програмного забезпечення

На цьому етапі здійснюється проєктування та кодування програмного забезпечення (ПЗ) для наземної станції управління (НСУ) та бортового контролера безпілотного літального апарату (БПЛА). Розроблене ПЗ повинно реалізовувати функціональність багатопортового зв'язку, керування динамічною зміною точок трансляції сигналу, а також забезпечувати необхідний рівень безпеки.

Цей етап умовно ділиться на два: розробка ПЗ для наземної станції управління та розробка ПЗ для БПЛА.

ПЗ НСУ є центральним елементом системи управління, що забезпечує оператору інтерфейс для керування БПЛА та координує роботу багатопортового каналу зв'язку та системи зміни точок трансляції.

Першою частиною ПЗ є інтерфейс оператора для керування БПЛА та моніторингу стану каналу зв'язку.

Основні вимоги:

- Відображення телеметричних даних (положення, висота, швидкість, стан батареї, рівень сигналу тощо) з кожного активного порту.
- Індикація поточної активної точки трансляції сигналу та списку доступних точок.
- Візуалізація стану кожного порту (активний/неактивний, рівень сигналу, рівень завад).
- Можливість ручного переключення між доступними точками трансляції сигналу.

- Елементи керування польотом БПЛА (джойстики, кнопки, сенсорні елементи).
- Відображення попереджень та сповіщень про стан каналу зв'язку та спроби несанкціонованого втручання.
- Модуль для налаштування параметрів багатопортового зв'язку та критеріїв автоматичного переключення точок трансляції.
- Функціональність запису та аналізу телеметричних даних та подій, пов'язаних зі станом каналу зв'язку.

Наступною частиною є модуль керування багатопортовим зв'язком. Основні вимоги:

- Агрегація даних – розподіл команд управління та телеметричних даних на окремі потоки для передачі через різні порти відповідно до розроблених алгоритмів (дублювання, розподіл за типом даних, чергування).
- Керування передачею - відправлення сформованих пакетів даних через відповідні радіоінтерфейси, пов'язані з кожним портом.
- Прийом даних
- Дезагрегація даних – об'єднання отриманих з різних портів фрагментів даних у вихідний потік команд управління або телеметрії.
- Контроль якості зв'язку – моніторинг рівня сигналу, втрати пакетів, затримок на кожному порту для прийняття рішень щодо перерозподілу трафіку або ініціації переключення точки трансляції.

Наступною частиною ПЗ є модуль керування точками трансляції. Він є ключовим компонентом програмного забезпечення відповідаючи за ефективне використання доступних ретрансляторів сигналу для забезпечення стабільного та захищеного зв'язку. Його функціональність починається з керування списком доступних точок трансляції, де зберігається вся необхідна інформація про кожну потенційну точку, включаючи її унікальний ідентифікатор, точні географічні координати, підтримувані частотні діапазони та протоколи зв'язку.

На основі цієї інформації реалізуються алгоритми автоматичного вибору оптимальної точки трансляції, які враховують ряд критично важливих критеріїв,

таких як поточний рівень сигналу, виміряний рівень радіоелектронних завад, прогнозована загроза РЕБ на основі аналізу електромагнітного спектра або розвідувальних даних, а також загальна доступність каналу зв'язку через цю точку. Після визначення оптимальної точки, модуль ініціює процес переключення, генеруючи та відправляючи відповідні команди як на БПЛА, так і, за необхідності, на обладнання самої точки трансляції для здійснення зміни активного каналу зв'язку. Важливою функцією є постійний моніторинг стану всіх доступних точок трансляції, що включає отримання інформації про їхню працездатність, поточну доступність для встановлення зв'язку та будь-які виявлені проблеми. Для забезпечення безперервного та узгодженого керування, модуль також реалізує протоколи синхронізації переключення, що передбачають обмін службовими повідомленнями між НСУ та БПЛА, гарантуючи одночасну зміну активної точки зв'язку обома сторонами та мінімізуючи час переривання зв'язку.

Наступним є реалізація механізмів виявлення та протидії спробам несанкціонованого втручання. Ці механізми включають безперервний аналіз трафіку даних на предмет аномалій, таких як поява незвичайних пакетів, несанкціоновані зміни частоти передачі або виявлення спроб підключення з невідомих джерел, що може свідчити про спроби перехоплення або підміни даних.

Одночасно здійснюється активне виявлення спроб глушіння на різних радіочастотних діапазонах, що використовуються для зв'язку, шляхом моніторингу рівня сигналу та спектральної щільності потужності. У випадку виявлення підозрілої активності або підтвердженої загрози, система автоматично реагує відповідно до заздалегідь визначених алгоритмів, наприклад, ініціюючи автоматичне переключення на інший доступний порт зв'язку або резервну точку трансляції сигналу для уникнення впливу загрози. Крім того, система генерує відповідні сповіщення для оператора наземної станції управління, інформуючи його про виявлену загрозу та вжиті автоматичні заходи, надаючи можливість для подальшого ручного втручання або аналізу ситуації.

Другою фундаментальною частиною є розробка бортового ПЗ для БПЛА. Бортове ПЗ відповідає за керування радіозв'язком, обробку команд управління, передачу телеметрії та взаємодію з системою динамічної зміни точок трансляції.

Першим модулем є модуль керування підключенням до точок трансляції. Він відповідає за встановлення та підтримку зв'язку з ретрансляторами сигналу, необхідними для передачі команд управління та телеметричних даних. Його робота починається з активного пошуку доступних точок трансляції шляхом сканування визначених радіочастотних діапазонів з метою виявлення сигналів, що ідентифікуються як вихідні від легітимних ретрансляторів. Після виявлення потенційної точки, модуль ініціює процедуру встановлення з'єднання, яка може включати обмін службовими повідомленнями для аутентифікації та узгодження параметрів зв'язку. Протягом польоту модуль постійно перебуває в режимі очікування та обробки команд на переключення, що надходять від наземної станції управління (НСУ), виконуючи інструкції щодо зміни поточної активної точки трансляції. Для забезпечення безперебійного керування та уникнення втрати зв'язку під час переключення, модуль також реалізує протоколи синхронізації переключення, здійснюючи обмін службовими повідомленнями з НСУ для узгодження моменту зміни активної точки обома сторонами, що гарантує плавний та контрольований перехід між різними ретрансляторами.

Другим є модуль керування багатопортовим зв'язком, який відповідає за ефективну обробку даних, що надходять та передаються через кілька активних радіочастотних каналів. Його функціональність включає дезагрегацію команд управління, де отримані з різних активних портів пакети даних об'єднуються у єдиний вихідний потік команд, зрозумілий для системи управління польотом БПЛА.

Одночасно модуль здійснює агрегацію телеметричних даних, розподіляючи різноманітну інформацію про стан апарату та параметри польоту на окремі потоки, призначені для передачі через різні порти. Керування передачею телеметрії полягає у відправленні цих сформованих пакетів через відповідні радіоінтерфейси, забезпечуючи доставку інформації на наземну

станцію управління. У зворотному напрямку модуль відповідає за прийом пакетів команд управління, що надходять з різних портів.

Важливою функцією є постійний контроль якості зв'язку на кожному активному порту, що включає моніторинг рівня прийнятого сигналу, виявлення втрати пакетів даних та вимірювання затримок передачі. Отримана інформація про якість зв'язку використовується для інформування наземної станції управління про поточний стан каналів та може слугувати основою для прийняття бортовим ПЗ рішень щодо ініціації перепідключення або використання альтернативних портів у випадку погіршення зв'язку.

2.4 Інтеграція та тестування

На цьому етапі відбувається об'єднання розроблених апаратних та програмних компонентів у єдину функціональну систему та проводяться всебічні випробування для підтвердження її працездатності, відповідності вимогам безпеки та продуктивності.

Першим кроком є апаратна інтеграція. На даному етапі здійснюється фізичне об'єднання всіх складових системи, що включає: встановлення та підключення радіопередавачів і приймачів з підтримкою багатопортовості до обчислювальних платформ наземної станції управління (НСУ) та бортового контролера БПЛА, забезпечуючи коректність підключення та електроживлення. Також проводиться підключення та налаштування відповідних антенних систем (спрямованих та всеспрямованих) до радіомодулів на НСУ, БПЛА та точках трансляції, з необхідним налаштуванням орієнтації спрямованих антен та перевіркою їхньої працездатності.

Важливим аспектом є інтеграція розробленого модуля керування зв'язком з існуючими системами управління польотом БПЛА для забезпечення передачі команд управління та отримання телеметричних даних. Крім того, здійснюється фізичне розгортання обладнання точок трансляції в запланованих місцях, їх підключення до джерел живлення та налаштування радіопараметрів, таких як

частоти, потужність та протоколи зв'язку, з обов'язковою перевіркою їхньої готовності до роботи.

На етапі програмної інтеграції відбувається об'єднання всіх розроблених програмних модулів та налагодження їхньої злагодженої взаємодії. Це включає встановлення та конфігурацію створеного програмного забезпечення на відповідних операційних системах наземної станції управління і бортового програмного забезпечення БПЛА, з обов'язковою перевіркою сумісності та відсутності конфліктів з іншим встановленим програмним забезпеченням. Ключовим аспектом є тестування взаємодії між модулями багатопортового зв'язку та керування точками трансляції, що передбачає перевірку коректної передачі даних між модулями агрегації та дезагрегації, керування портами зв'язку та керування точками трансляції на обох кінцях каналу зв'язку – як на НСУ, так і на БПЛА. Також проводиться ретельна перевірка коректності функціонування інтегрованих криптографічних алгоритмів та протоколів безпеки, включаючи тестування процесів шифрування та дешифрування даних, механізмів автентифікації та авторизації, а також оцінка стійкості застосованих криптографічних методів до відомих кібератак. На завершення цього під етапу здійснюється інтеграція інтерфейсу оператора з усіма модулями керування зв'язком, що включає перевірку коректного відображення всієї необхідної інформації про стан каналу зв'язку, доступні та активні точки трансляції, а також забезпечення повноцінної можливості оператора керувати всіма аспектами системи через розроблений графічний інтерфейс.

Тестування системи включає в себе всебічні випробування інтегрованої системи для оцінки її функціональності, продуктивності та безпеки.

Першим на черзі є функціональне тестування. Воно включає в себе:

- Перевірка можливості встановлення та підтримки багатопортового зв'язку між НСУ та БПЛА.

- Тестування передачі команд управління та отримання телеметричних даних через різні порти.

- Перевірка ініціації та успішного виконання переключення між доступними точками трансляції сигналу (як автоматичного, так і за командою оператора).

- Тестування протоколів синхронізації переключення між НСУ та БПЛА.

- Перевірка роботи механізмів виявлення спроб несанкціонованого втручання та відповідного реагування системи.

Наступним є тестування продуктивності, що включає в себе:

- Вимірювання затримок передачі даних в каналі управління при використанні багатопортовості.

- Оцінка пропускну здатності каналу зв'язку при різних конфігураціях портів.

- Вимірювання часу, необхідного для переключення між точками трансляції.

- Оцінка стабільності зв'язку на різних відстанях та в різних умовах радіоефіру.

Наступним є тестування в безпеці, що включає в себе:

- Імітація різних типів радіоелектронних завад (шумові, імпульсні, обманні) для оцінки стійкості каналу управління.

- Перевірка стійкості до спроб перехоплення та дешифрування переданих даних.

- Імітація спроб несанкціонованого втручання та підміни команд управління.

- Тестування стійкості до атак типу "людина посередині".

- Оцінка ефективності механізмів автентифікації та авторизації.

Останнім є стрес-тестування, під час якого проводиться:

- Перевірка стабільності роботи системи при високому навантаженні (максимальна кількість одночасно активних портів, інтенсивний обмін даними).

- Тестування в умовах сильних радіоперешкод та при спробах одночасного впливу на кілька частотних діапазонів.

- Оцінка поведінки системи при виникненні нештатних ситуацій (відмова одного з портів, втрата зв'язку з точкою трансляції).

2.5 Оцінка результатів

На цьому завершальному етапі проводиться аналіз результатів проведеного тестування, виявляються недоліки та вузькі місця розробленої системи, здійснюється її оптимізація та, за необхідності, проводяться повторні випробування для досягнення заданих вимог.

На етапі аналізу результатів тестування здійснюється ретельний розгляд усіх зібраних даних, отриманих на попередньому етапі випробувань. Це включає детальну оцінку ефективності розробленої системи за попередньо визначеними критеріями, де фактичні показники роботи порівнюються з вимогами безпеки та продуктивності, сформульованими на етапі аналізу загроз. Зокрема, оцінюється стійкість каналу зв'язку до різних видів радіоелектронного придушення, ефективність заходів з маскування місцезнаходження наземної станції управління, рівень забезпеченої конфіденційності переданих даних, надійність механізмів автентифікації та гарантування цілісності інформації, оперативність процесу переключення між точками трансляції сигналу, включаючи час переключення та можливі переривання зв'язку, величини затримок у каналі управління, а також енергоефективність використаних технічних рішень.

Проводиться детальний аналіз записаних під час тестування журналів подій та телеметричних даних з метою виявлення будь-яких помилок у роботі системи, неочікуваної поведінки, проблем зі встановленням або підтримкою зв'язку, зафіксованих спроб несанкціонованого втручання та адекватності реакції системи на такі події. Аналіз телеметричних даних дозволяє отримати об'єктивну картину стабільності зв'язку, рівня сигналу на різних активних портах та точках трансляції, а також оцінити ефективність застосованих алгоритмів вибору оптимальних точок та процедур переключення між ними.

На завершення цього етапу здійснюється збір зворотного зв'язку від операторів, які безпосередньо брали участь у проведенні випробувань, щодо їхньої суб'єктивної оцінки зручності використання розробленого інтерфейсу, ефективності системи управління багатопортовим зв'язком та динамічною зміною точок трансляції, а також їхнього загального враження від функціонування системи в умовах тестування.

На основі проведеного аналізу виявляються проблемні аспекти розробленої системи. Проводиться ідентифікація факторів, що обмежують пропускну здатність каналу, збільшують затримки, знижують дальність зв'язку або призводять до нестабільності роботи.

Наступними виявляються вразливості безпеки, а саме: ідентифікація слабких місць у реалізації криптографічних алгоритмів, протоколів автентифікації, механізмів виявлення та протидії втручанню, які можуть бути використані зловмисниками.

Останнім проводиться виявлення проблем з функціональністю: ідентифікація некоректної роботи окремих модулів або функцій системи, наприклад, неправильний вибір точки трансляції, нестабільне переключення між портами, помилки в обробці даних тощо.

На основі проведеного аналізу та виявлених недоліків вносяться необхідні зміни до апаратного та програмного забезпечення.

2.6 Висновки до другого розділу

У другому розділі було детально описано методологію створення захищеного багатопортового каналу управління БПЛА з динамічною зміною точок трансляції сигналу.

Розділ охоплює п'ять основних етапів процесу розробки, починаючи з аналізу потенційних загроз та визначення ключових вимог безпеки до системи.

Далі, було представлено етап проєктування багатопортового каналу зв'язку, де розглядаються важливі аспекти вибору фізичного рівня зв'язку, радіочастотних

діапазонів, модуляційних схем, типів антен та архітектури багатопортовості. Також, було приділено увагу розробці алгоритмів розподілу даних між портами та проєктуванню системи динамічної зміни точок трансляції, включаючи критерії вибору оптимальних точок та протоколи переключення між ними.

Наступні етапи описують розробку програмного забезпечення для наземної станції управління та бортового контролера БПЛА, деталізуючи вимоги до інтерфейсу оператора, модулів керування багатопортовістю та точками трансляції, а також механізмів виявлення та протидії спробам несанкціонованого втручання.

Завершальні етапи включають опис процесу інтеграції та тестування розробленої системи, а також методологію оцінки отриманих результатів, виявлення недоліків та оптимізації системи.

Таким чином, другий розділ надає інформацію про практичні кроки та технічні рішення, необхідні для створення ефективного та безпечного каналу управління БПЛА.

РОЗДІЛ III. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАХИЩЕНОГО КАНАЛУ

Для створення захищеного каналу необхідно розробити програмне забезпечення. Воно складається з 3 основних частин: код для управління ретрансляторами наземної станції, код для управління ретрансляторами, що безпосередньо передають сигнали управління БПЛА, код для зміни точок ретрансляції.

3.1 Код для управління ретрансляторами наземної станції

Основні вимоги до коду управління ретрансляторами для наземної станції управління (НСУ) БПЛА:

- Керує кількома радіоканалами (портами) одночасно
- Автоматично переключається між точками трансляції при погіршенні якості зв'язку
- Використовує шифрування (AES-256-CBC) та автентифікацію (HMAC-SHA256)
- Відправляє команди через активні канали для підвищення надійності

```

10 class SecureMultiPortChannel:
11     def __init__(self):
12         # Конфігурація каналів
13         self.ports = {
14             'primary': {'freq': 2400, 'status': 'active', 'socket': None, 'key': b'32_byte_key_for_AES_256_CBC_mode'},
15             'secondary1': {'freq': 5800, 'status': 'standby', 'socket': None, 'key': b'another_32_byte_key_for_AES_'},
16             'secondary2': {'freq': 900, 'status': 'standby', 'socket': None, 'key': b'yet_another_32_byte_key_here'}
17         }
18
19         # Список доступних точок трансляції
20         self.relay_points = [
21             {'id': 1, 'location': (50.45, 30.52), 'status': 'active', 'freqs': [2400, 5800]},
22             {'id': 2, 'location': (50.46, 30.53), 'status': 'standby', 'freqs': [900, 2400]},
23             {'id': 3, 'location': (50.44, 30.51), 'status': 'standby', 'freqs': [5800, 900]}
24         ]
25
26         # Параметри зв'язку
27         self.current_relay = 1
28         self.switch_threshold = -70 # dBm
29         self.encryption_enabled = True
30         self.auth_enabled = True
31
32         # Статистика
33         self.link_quality = {'primary': 0, 'secondary1': 0, 'secondary2': 0}
34         self.packet_loss = {'primary': 0, 'secondary1': 0, 'secondary2': 0}
35
36         # Ініціалізація сокетів
37         self._init_sockets()
38
39         # Запуск моніторингу якості зв'язку
40         self.monitor_thread = threading.Thread(target=self._monitor_link_quality, daemon=True)
41         self.monitor_thread.start()
42
43     def _init_sockets(self):
44         """Ініціалізація сокетів для кожного порту"""
45         for port in self.ports.values():
46             try:
47                 # Тут має бути реальна ініціалізація радіоінтерфейсу
48                 # Для прикладу використовуємо UDP сокети
49                 port['socket'] = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
50                 port['socket'].settimeout(0.1)
51             except Exception as e:
52                 print(f"Помилка ініціалізації порту {port['freq']}MHz: {e}")
53
54     def _monitor_link_quality(self):
55         """Функція моніторингу якості зв'язку"""
56         while True:
57             for port_name, port in self.ports.items():
58                 if port['status'] == 'active':
59                     # Імітація вимірювання якості зв'язку
60                     self.link_quality[port_name] = random.randint(-90, -50)
61                     self.packet_loss[port_name] = random.random() * 10
62
63                     # Перевірка на необхідність переключення
64                     if self.link_quality[port_name] < self.switch_threshold:
65                         print(f"Погана якість зв'язку на {port_name} ({self.link_quality[port_name]}dBm). Спробуємо переключитися...")
66                         self._switch_relay_point()
67
68         time.sleep(1)

```

Рис. 3.1 Код для НСУ – встановлення каналів

Даний код описує основний клас SecureMultiPortChannel, який інкапсулює логіку багатопортового зв'язку, шифрування, автентифікації та перемикавання між точками трансляції.

Канал використовує декілька портів зв'язку (primary, secondary1, secondary2) з різними частотами. Це дозволяє забезпечити резервування та підвищити стійкість до перешкод.

Код постійно моніторить якість зв'язку на активному порту та автоматично перемикається на інший порт або точку трансляції у разі погіршення якості.

```

70 def _switch_relay_point(self):
71     """Переключення на іншу точку трансляції"""
72     # Знаходимо доступні точки трансляції
73     available_relays = [rp for rp in self.relay_points
74                         if rp['status'] == 'standby' and rp['id'] != self.current_relay]
75
76     if not available_relays:
77         print("Немає доступних точок трансляції для переключення")
78         return
79
80     # Вибираємо нову точку трансляції (можна додати складніші критерії вибору)
81     new_relay = random.choice(available_relays)
82
83     # Формуємо команду переключення
84     switch_cmd = {
85         'type': 'relay_switch',
86         'new_relay_id': new_relay['id'],
87         'timestamp': time.time(),
88         'signature': ''
89     }
90
91     # Додаємо підпис (якщо автентифікація увімкнена)
92     if self.auth_enabled:
93         switch_cmd['signature'] = self._sign_data(str(switch_cmd).encode())
94
95     # Надсилаємо команду через всі активні порти
96     for port_name, port in self.ports.items():
97         if port['status'] == 'active':
98             try:
99                 encrypted_cmd = self._encrypt_data(str(switch_cmd).encode(), port['key'])
100                 # Тут має бути реальне відправлення на БПЛА
101                 # Для прикладу використовуємо UDP
102                 port['socket'].sendto(encrypted_cmd, ('drone_ip', 14550))
103                 print(f"Надіслано команду переключення на точку {new_relay['id']} через {port_name}")
104             except Exception as e:
105                 print(f"Помилка відправлення команди через {port_name}: {e}")
106
107     # Оновлюємо поточну точку трансляції
108     for rp in self.relay_points:
109         if rp['id'] == new_relay['id']:
110             rp['status'] = 'active'
111         elif rp['id'] == self.current_relay:
112             rp['status'] = 'standby'
113
114     self.current_relay = new_relay['id']
115     print(f"Успішно переключено на точку трансляції {new_relay['id']}")
116

```

Рис. 3.2 Код для НСУ – вибір точки трансляції

Код може використовувати декілька точок трансляції для ретрансляції сигналу. Це ускладнює виявлення та придушення каналу зв'язку противником.

```

163     def _encrypt_data(self, data, key):
164         """Шифрування даних AES-256-CBC"""
165         if not self.encryption_enabled:
166             return data
167
168         # Генеруємо IV (Initialization Vector)
169         iv = np.random.bytes(16)
170
171         # Шифруємо дані
172         cipher = Cipher(
173             algorithms.AES(key),
174             modes.CBC(iv),
175             backend=default_backend()
176         )
177         encryptor = cipher.encryptor()
178
179         # Доповнюємо дані до кратності 16 байт
180         pad_length = 16 - (len(data) % 16)
181         data += bytes([pad_length]) * pad_length
182
183         encrypted = encryptor.update(data) + encryptor.finalize()
184         return iv + encrypted
185
186     def _decrypt_data(self, encrypted_data, key):
187         """Дешифрування даних AES-256-CBC"""
188         if not self.encryption_enabled:
189             return encrypted_data
190
191         if len(encrypted_data) < 16:
192             raise ValueError("Зашифровані дані занадто короткі")
193
194         iv = encrypted_data[:16]
195         cipher = Cipher(
196             algorithms.AES(key),
197             modes.CBC(iv),
198             backend=default_backend()
199         )
200         decryptor = cipher.decryptor()
201
202         decrypted = decryptor.update(encrypted_data[16:]) + decryptor.finalize()
203

```

Рис. 3.3 Код для НСУ – шифрування та підпис

Для шифрування даних використовується алгоритм AES-256-CBC. Це забезпечує конфіденційність переданих даних.

Для автентифікації даних використовуються HMAC-підписи. Це дозволяє переконатися в цілісності та автентичності повідомлень.

Класи та методи коду:

- `__init__(self)`: Конструктор класу, ініціалізує параметри каналу, порти, точки трансляції, запускає моніторинг.
- `__init__sockets(self)`: Створює UDP сокети для кожного порту.

- `_monitor_link_quality(self)`: Моніторить якість зв'язку та ініціює переключення при потребі.

- `_switch_relay_point(self)`: Перемикає на іншу точку трансляції.

- `_send_command(self, command)`: Надсилає команду на БПЛА.

- `_handle_port_failure(self, port_name)`: Обробляє збій порту.

- `_encrypt_data(self, data, key)`: Шифрує дані за допомогою AES-256-CBC.

- `_decrypt_data(self, encrypted_data, key)`: Дешифрує дані.

- `_sign_data(self, data)`: Створює HMAC-підпис для даних.

- `_verify_signature(self, data, signature)`: Перевіряє HMAC-підпис.

- `_receive_telemetry(self)`: Отримує телеметрію від БПЛА.

3.2 Код для управління ретрансляторами, що безпосередньо передають сигнали управління БПЛА

Основні вимоги до коду для БПЛА:

- Отримує команди через активний радіоканал
- Автоматично переключається між точками ретрансляції за командою з наземної станції
- Відправляє телеметричні дані через всі активні канали
- Використовує такі ж механізми шифрування та автентифікації, як і наземна станція

```

class DroneCommunication:
    def __init__(self):
        # Конфігурація каналів (ключі мають відповідати наземній станції)
        self.ports = {
            'primary': {'freq': 2400, 'status': 'active', 'socket': None, 'key': b'32_byte_key_for_AES_256_CBC_mode'},
            'secondary1': {'freq': 5800, 'status': 'standby', 'socket': None, 'key': b'another_32_byte_key_for_AES_'},
            'secondary2': {'freq': 900, 'status': 'standby', 'socket': None, 'key': b'yet_another_32_byte_key_here'}
        }

        # Поточна точка ретрансляції
        self.current_relay = 1

        # Параметри безпеки
        self.encryption_enabled = True
        self.auth_enabled = True

        # Черга команд
        self.command_queue = []
        self.command_lock = threading.Lock()

        # Ініціалізація сокетів
        self._init_sockets()

        # Запуск потоку для отримання команд
        self.receiver_thread = threading.Thread(target=self._receive_commands, daemon=True)
        self.receiver_thread.start()

        # Запуск потоку для відправлення телеметрії
        self.telemetry_thread = threading.Thread(target=self._send_telemetry, daemon=True)
        self.telemetry_thread.start()

    def _init_sockets(self):
        """Ініціалізація сокетів для кожного порту"""
        for port in self.ports.values():
            try:
                # Тут має бути реальна ініціалізація радіоінтерфейсу
                # Для прикладу використовуємо UDP сокети
                port['socket'] = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
                port['socket'].bind(('0.0.0.0', 14550))
                port['socket'].settimeout(0.1)
            except Exception as e:
                print(f"Помилка ініціалізації порту {port['freq']}MHz: {e}")

```

Рис. 3.4 Код для БПЛА – вибір каналів

Даний код описує основний клас DroneCommunication, який інкапсулює логіку багатопортового зв'язку, шифрування, автентифікації та перемикання між точками трансляції.

Визначає кілька каналів зв'язку (primary, secondary1, secondary2) з різними частотами, статусами (активний, в режимі очікування) та криптографічними ключами.

Створює та прив'язує UDP сокети для кожного визначеного каналу зв'язку, а також встановлює тайм-аут для операцій з сокетами.

```

def _receive_commands(self):
    """Отримання команд від наземної станції"""
    while True:
        for port_name, port in self.ports.items():
            if port['status'] == 'active':
                try:
                    data, addr = port['socket'].recvfrom(4096)
                    if data:
                        decrypted = self._decrypt_data(data, port['key'])
                        command = eval(decrypted.decode()) # У реальному коді використовуйте безпечний парсер

                        # Перевіряємо підпис (якщо автентифікація увімкнена)
                        if self.auth_enabled and not self.verify_signature(
                            str(command['command']).encode(),
                            command['signature']
                        ):
                            print(f"Недійсний підпис команди з {port_name}!")
                            continue

                        print(f"Отримано команду через {port_name}: {command['command']}")

                        # Обробка команди переключення ретранслятора
                        if command['type'] == 'relay_switch':
                            self.current_relay = command['new_relay_id']
                            print(f"Переключено на точку ретрансляції {self.current_relay}")
                        else:
                            # Додаємо команду до черги
                            with self.command_lock:
                                self.command_queue.append(command['command'])
                except socket.timeout:
                    continue
                except Exception as e:
                    print(f"Помилка отримання команди через {port_name}: {e}")

            time.sleep(0.01)

def get_next_command(self):
    """Отримання наступної команди з черги"""
    with self.command_lock:
        if self.command_queue:
            return self.command_queue.pop(0)
    return None

```

Рис. 3.5 Код для БПЛА – команда для зміни точки ретрансляції

Отримання команд за допомогою `_receive_commands` Запускається в окремому потоці для неблокуючого отримання даних. При отриманні даних на активному каналі:

- Дешифрує дані
- Парсить отримані дані як команду Python
- Перевіряє підпис команди
- Безпечно витягує та видаляє першу команду з черги `command_queue`, використовуючи блокування для запобігання гонкам.

```

126 def _get_current_telemetry(self):
127     """Отримання поточної телеметрії (заглушка)"""
128     # У реальній системі це має бути зчитування з датчиків
129     return {
130         'altitude': random.uniform(0, 100),
131         'latitude': 50.45 + random.uniform(-0.01, 0.01),
132         'longitude': 30.52 + random.uniform(-0.01, 0.01),
133         'speed': random.uniform(0, 20),
134         'battery': random.uniform(20, 100),
135         'status': 'OK',
136         'relay_point': self.current_relay
137     }
138
139 def _encrypt_data(self, data, key):
140     """Шифрування даних AES-256-CBC (аналогічно наземній станції)"""
141     if not self.encryption_enabled:
142         return data
143
144     iv = np.random.bytes(16)
145     cipher = Cipher(
146         algorithms.AES(key),
147         modes.CBC(iv),
148         backend=default_backend()
149     )
150     encryptor = cipher.encryptor()
151     pad_length = 16 - (len(data) % 16)
152     data += bytes([pad_length]) * pad_length
153     encrypted = encryptor.update(data) + encryptor.finalize()
154     return iv + encrypted
155
156 def _decrypt_data(self, encrypted_data, key):
157     """Дешифрування даних AES-256-CBC (аналогічно наземній станції)"""
158     if not self.encryption_enabled:
159         return encrypted_data
160
161     if len(encrypted_data) < 16:
162         raise ValueError("Зашифровані дані занадто короткі")
163
164     iv = encrypted_data[:16]
165     cipher = Cipher(
166         algorithms.AES(key),
167         modes.CBC(iv),
168         backend=default_backend()
169     )
170     decryptor = cipher.decryptor()
171     decrypted = decryptor.update(encrypted_data[16:]) + decryptor.finalize()
172     pad_length = decrypted[-1]
173     return decrypted[:-pad_length]
174
175 def _sign_data(self, data):
176     """Створення HMAC-підпису (аналогічно наземній станції)"""
177     if not self.auth_enabled:
178         return b''
179
180     h = hmac.HMAC(b'shared_secret_key_for_hmac', hashes.SHA256(), backend=default_backend())
181     h.update(data)
182     return h.finalize()
183

```

Рис. 3.6 Код для БПЛА – шифрування та підпис

Шифрування даних (`_encrypt_data`) реалізує шифрування даних за допомогою алгоритму AES-256 в режимі CBC. Використовує випадковий вектор ініціалізації, який додається до зашифрованих даних. Застосовує PKCS#7 padding для вирівнювання довжини даних до кратного розміру блоку AES.

Дешифрування даних (`_decrypt_data`) реалізує дешифрування даних, зашифрованих за допомогою `_encrypt_data`. Витягує з початку зашифрованих даних. Видаляє padding після дешифрування.

Створення підпису (`_sign_data`) створює HMAC-SHA256 підпис для забезпечення автентичності даних. Використовує спільний секретний ключ (`b'shared_secret_key_for_hmac'`).

Перевірка підпису (`_verify_signature`) перевіряє HMAC-SHA256 підпис отриманих даних.

3.3 Код для точки ретрансляції

Основні вимоги до коду для точок ретрансляції:

- Ретрансляція даних між наземною станцією та БПЛА
- Може працювати з різними частотами

```

7
8 class RelayPoint:
9     def __init__(self, relay_id, location, supported_freqs):
10         self.relay_id = relay_id
11         self.location = location
12         self.supported_freqs = supported_freqs
13         self.status = 'standby'
14
15         # Сокети для кожного підтримуваного частотного діапазону
16         self.sockets = {}
17         for freq in supported_freqs:
18             try:
19                 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
20                 sock.bind(('0.0.0.0', 14550 + freq//100)) # Унікальний порт для кожної частоти
21                 sock.settimeout(0.1)
22                 self.sockets[freq] = sock
23             except Exception as e:
24                 print(f"Помилка ініціалізації сокету для частоти {freq}MHz: {e}")
25
26         # Запуск потоку для ретрансляції даних
27         self.relay_thread = threading.Thread(target=self._relay_data, daemon=True)
28         self.relay_thread.start()
29
30     def _relay_data(self):
31         """Ретрансляція даних між наземною станцією та БПЛА"""
32         while True:
33             for freq, sock in self.sockets.items():
34                 try:
35                     # Отримуємо дані від наземної станції або БПЛА
36                     data, addr = sock.recvfrom(4096)
37                     if data:
38                         # Визначаємо напрямок передачі (спрощено)
39                         if addr[0].startswith('192.168'): # Приклад - дані від наземної станції
40                             target = ('drone_ip', 14550)
41                         else: # Дані від БПЛА
42                             target = ('ground_station_ip', 14550)
43
44                         # Ретранслюємо дані
45                         sock.sendto(data, target)
46                         print(f"Ретрансльовано {len(data)} байт на частоті {freq}MHz")
47                     except socket.timeout:
48                         continue
49                     except Exception as e:
50                         print(f"Помилка ретрансляції на частоті {freq}MHz: {e}")
51
52                 time.sleep(0.01)
53
54     def activate(self):
55         """Активізація точки ретрансляції"""
56         self.status = 'active'
57         print(f"Точка ретрансляції {self.relay_id} активована")
58
59     def deactivate(self):
60         """Деактивація точки ретрансляції"""
61         self.status = 'standby'
62         print(f"Точка ретрансляції {self.relay_id} деактивована")
63

```

Рис. 3.7 Код для точок ретрансляції

Клас RelayPoint ініціалізує основні атрибути точки ретрансляції:

- relay_id: Унікальний ідентифікатор точки ретрансляції.
- location: Координати розташування точки ретрансляції (широта, довгота).

- `supported_freqs`: Список частотних діапазонів, які підтримує дана точка ретрансляції для зв'язку.

- `status`: Поточний статус точки ретрансляції ('standby' за замовчуванням).

Створює словник `self.sockets` для зберігання сокетів, прив'язаних до кожної підтримуваної частоти.

Для кожної підтримуваної частоти:

- Створює UDP сокет (`socket.AF_INET`, `socket.SOCK_DGRAM`).

- Прив'язує сокет до унікального порту на локальній машині ('0.0.0.0', $14550 + \text{freq}/100$). Використання $\text{freq}/100$ забезпечує відносно унікальний номер порту для кожної сотні мегагерц частоти.

- Встановлює невеликий тайм-аут для операцій з сокетом (`sock.settimeout(0.1)`).

- Зберігає створений сокет у словнику `self.sockets` з частотою як ключем.

- Обробляє можливі винятки під час ініціалізації сокетів та виводить повідомлення про помилку.

Запускає окремий потік (`self.relay_thread`) для виконання методу `_relay_data` у фоновому режимі. Встановлює `daemon=True`, що означає, що потік завершиться, коли завершиться основний потік програми.

`_relay_data(self)` – метод, який відповідає за ретрансляцію даних між наземною станцією та БПЛА.

3.4 Висновки до третього розділу

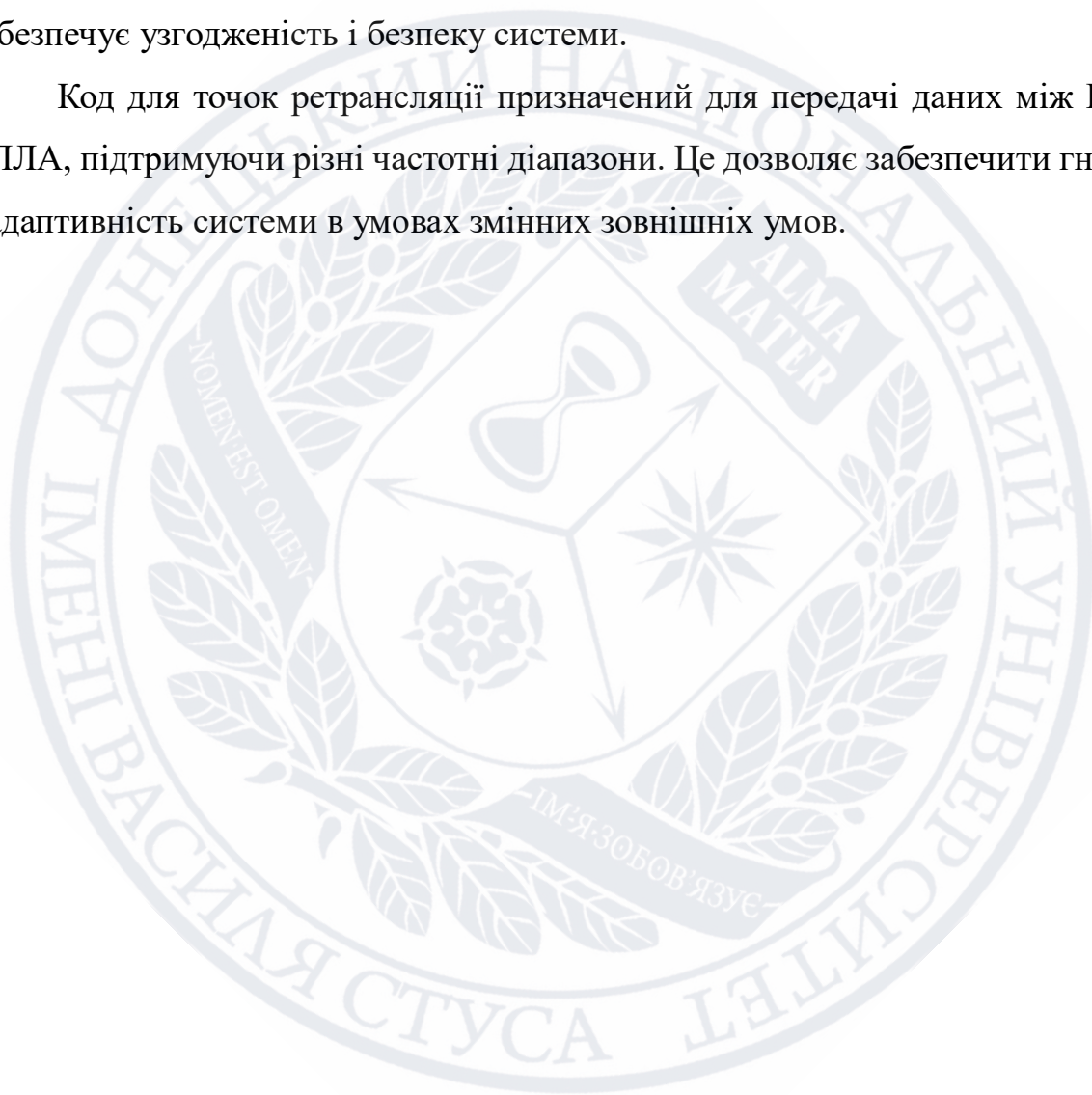
У третьому розділі було детально розглянуто програмне забезпечення, необхідне для реалізації захищеного багатопортового каналу управління БПЛА. Основна увага приділялася розробці коду для трьох ключових компонентів системи: наземної станції управління (НСУ), безпілотного літального апарату (БПЛА) та точок ретрансляції.

Код для НСУ забезпечує керування кількома радіоканалами, автоматичне переключення між точками трансляції при погіршенні якості зв'язку, а також

використовує сучасні методи шифрування (AES-256-CBC) та автентифікації (HMAC-SHA256). Це дозволяє підвищити надійність, конфіденційність і цілісність переданих даних.

Код для БПЛА реалізує функціонал отримання команд через активний радіоканал, автоматичного переключення між точками ретрансляції за командою з НСУ, а також відправлення телеметричних даних через всі активні канали. Використання однакових механізмів шифрування та автентифікації, як і на НСУ, забезпечує узгодженість і безпеку системи.

Код для точок ретрансляції призначений для передачі даних між НСУ та БПЛА, підтримуючи різні частотні діапазони. Це дозволяє забезпечити гнучкість і адаптивність системи в умовах змінних зовнішніх умов.



ВИСНОВКИ

У ході виконання бакалаврської роботи було досягнуто поставленої мети – розроблено концепцію захищеного багатопортового каналу зв'язку для управління безпілотними літальними апаратами з динамічною зміною точок трансляції сигналу.

Було проведено детальний аналіз архітектури FPV-дронів, їхнього апаратного та програмного забезпечення, частотних діапазонів і протоколів зв'язку, таких як PX4, MAVLink, UART, UDP, SPI та RTPS. Обґрунтовано доцільність використання багатопортовості як засобу підвищення надійності, стійкості до радіоелектронних завад та безпеки управління.

У результаті розроблено програмне забезпечення для наземної станції управління та бортового комп'ютера БПЛА, яке забезпечує динамічне переключення між точками трансляції сигналу, використання кількох каналів зв'язку, шифрування даних (AES-256-CBC) та автентифікацію (HMAC-SHA256).

Таким чином, створено код для системи управління БПЛА, яка відповідає сучасним вимогам безпеки та може бути адаптована для подальшого вдосконалення та практичного використання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1.HE, Changxiao. The Social Application and Exploration of FPV Drone. World Scientific Research Journal, 2024, 10.6: 95-99. Режим доступу: <https://www.wsrj.org/download/WSRJ-10-6-95-99.pdf> (Дата звернення: 24.04.2025р.)
- 2.ŠKILJO, Maja; UDŽENIJA, Marina; BLAŽEVIĆ, Zoran. Axial Ratio of Quarter-and Half-wavelength Cloverleaf Antenna for Drone FPV Applications. In: 2023 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). IEEE, 2023. p. 1-5.Режим доступу: https://www.researchgate.net/profile/Zoran-Blazevic/publication/374593105_Axial_Ratio_of_Quarter-_and_Half-wavelength_Cloverleaf_Antenna_for_Drone_FPV_Applications/links/655fa8afb86a1d521b02f9e8/Axial-Ratio-of-Quarter-and-Half-wavelength-Cloverleaf-Antenna-for-Drone-FPV-Applications.pdf (Дата звернення: 20.04.2025р.)
- 3.SARATH, J. V., et al. REVIEW OF ANTENNAS USED IN FPV/WLAN APPLICATIONS. *Acta Technica Corviniensis-Bulletin of Engineering*, 2021, 14.1. Режим доступу: <https://acta.fih.upt.ro/pdf/2021-1/ACTA-2021-1-05.pdf> (Дата звернення: 24.04.2025р.)
- 4.Квадрокоптер DJI Mavic 3E Режим доступу: [Електронний ресурс] <https://www.optics-pro.com.ua/ua/kvadrokopteri/kvadrokopter-dji-mavic-3e> (Дата звернення: 24.04.2025р.)
- 5.Квадрокоптер DJI Mavic 3T Enterprise с тепловізором. Режим доступу: [Електронний ресурс] <https://drony.org.ua/dji-mavic-3t> (Дата звернення: 24.04.2025р.)
- 6.DJI Mavic 2 Enterprise Advanced. Режим доступу: [Електронний ресурс] <https://www.dji.com/support/product/mavic-2-enterprise-advanced> (Дата звернення: 24.04.2025р.)
- 7.MOLLICA, Christian M. FPV Flight Dynamics: *FPV Flight Dynamics: Mastering Acro Mode on High-Performance Drones*. Vespula Ventures LLC, 2020. Режим доступу: <https://books.google.com.ua/books?hl=uk&lr=&id=u60mEAAAQBAJ&oi=fnd&pg=>

[PA11&dq=MOLLICA,+Christian+M.+FPV+Flight+Dynamics:+Mastering+Acro+Mode+on+High-Performance+Drones.+Vespula+Ventures+LLC,+2020.&ots=lq6hJX6VI2&sig=9nnafDDXW-3OHT9khq9Irtujd8c&redir_esc=y#v=onepage&q=MOLLICA%2C%20Christian%20M.%20FPV%20Flight%20Dynamics%3A%20Mastering%20Acro%20Mode%20on%20High-Performance%20Drones.%20Vespula%20Ventures%20LLC%2C%202020.&f=false](#)
(Дата звернення: 24.04.2025р.)

8.DELMERICICO, Jeffrey, et al. Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset. In: 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019. p. 6713-6719. Режим доступу: https://www.zora.uzh.ch/id/eprint/197739/1/ICRA19_Delmerico.pdf (Дата звернення: 24.04.2025р.)

9.EFAZ, Erteza Tawsif, et al. Modeling of a high-speed and cost-effective FPV quadcopter for surveillance. In: 2020 23rd International Conference on Computer and Information Technology (ICCIT). IEEE, 2020. p. 1-6. Режим доступу: <https://ieeexplore.ieee.org/abstract/document/9392696> (Дата звернення: 24.04.2025р.)

10.Що таке проміжне програмне забезпечення? Режим доступу: [Електронний ресурс] <https://uk.theastrologypage.com/middleware> (Дата звернення: 24.04.2025р.)

11.MALACARNE, Francesco. PX4 autopilot customization for non-standard gimbal and UWB peripherals. 2020. PhD Thesis. Politecnico di Torino. Режим доступу: <https://webthesis.biblio.polito.it/secure/15919/1/tesi.pdf> (Дата звернення: 24.04.2025р.)

12.KUNZE, S.; WEINBERGER, A.; POESCHL, R. A software defined radio based implementation for the radio frequency analysis of signals from unmanned aerial systems. In: 2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA). IEEE, 2019. p. 1-6. Режим доступу:

https://www.researchgate.net/profile/Rainer-Poeschl/publication/344738029_A_Software_Defined_Radio_Based_Implementation_for_the_Radio_Frequency_Analysis_of_Signals_from_Unmanned_Aerial_Systems/links/600825b092851c13fe2404e0/A-Software-Defined-Radio-Based-Implementation-for-the-Radio-Frequency-Analysis-of-Signals-from-Unmanned-Aerial-Systems.pdf (Дата звернення: 24.04.2025р.)

13.UART протокол Режим доступу: [Електронний ресурс] <https://itmaster.biz.ua/directory/standarts/uart.html> (Дата звернення: 24.04.2025р.)

14.Communicating with the Quadcopter Режим доступу: [Електронний ресурс] <https://www.objc.io/issues/8-quadcopter/communicating-with-the-quadcopter/> (Дата звернення: 24.04.2025р.)

15.NASSI, Ben, et al. Game of drones-detecting streamed POI from encrypted FPV channel. arXiv preprint arXiv:1801.03074, 2018. Режим доступу: <https://arxiv.org/pdf/1801.03074> (Дата звернення: 24.04.2025р.)

16.WADE, Jackson. RC You Later: Development of a Remote-Controlled Hovercraft with First-Person View Capabilities. 2024. PhD Thesis. Middle Tennessee State University. Режим доступу: <https://jewlscholar.mtsu.edu/server/api/core/bitstreams/76715125-4e37-47ab-9ee1-f7e5c42dc9ff/content> (Дата звернення: 24.04.2025р.)

17.MILLSAP, Chris; GARVEY, Nathan; HAYAT, Faisal. Pilot-1: autonomous fixed-wing aircraft control system. 2015. Режим доступу: https://scholarcommons.scu.edu/cgi/viewcontent.cgi?article=1011&context=idp_senior (Дата звернення: 24.04.2025р.)

18.What is RTPS? Режим доступу: [Електронний ресурс] <https://www.eprosima.com/developer-resources/whitepapers/rtps> (Дата звернення: 24.04.2025р.)

19.JAVAID, Shumaila, et al. Communication and control in collaborative UAVs: Recent advances and future trends. IEEE Transactions on Intelligent Transportation Systems, 2023, 24.6: 5719-5739. Режим доступу: <https://arxiv.org/pdf/2302.12175> (Дата звернення: 24.04.2025р.)

20.CUI, Zhuangzhuang, et al. Channel modeling for UAV communications: State of the art, case studies, and future directions. arXiv preprint arXiv:2012.06707, 2020. Режим доступу: <https://arxiv.org/pdf/2012.06707> (Дата звернення: 24.04.2025р.)

21.ZHANG, Peiying, et al. UAV Communication in Space–Air–Ground Integrated Networks (SAGINs): Technologies, Applications, and Challenges. Drones (2504-446X), 2025, 9.2. Режим доступу: <https://www.mdpi.com/2504-446X/9/2/108> (Дата звернення: 24.04.2025р.)

22. SINEM, C., MUSTAFA, E., ANUJ, P., AHMAD, B. A Study of Channel Estimation in OFDM Systems. Режим доступу: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d8bc718b5d552404cee384975594665079cd7a3e> (Дата звернення: 10.05.2025р)

23.Boquet, Guillem, et al. "LR-FHSS: Overview and performance analysis." IEEE Communications Magazine 59.3 (2021): 30-36. Режим доступу: <https://arxiv.org/pdf/2010.00491> (Дата звернення: 10.05.2025р)

24. Liu, Yao, et al. "Randomized differential DSSS: Jamming-resistant wireless broadcast communication." 2010 Proceedings IEEE INFOCOM. IEEE, 2010. Режим доступу: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=115606e6c7c037054641c727728c0a42e358e3ed> Дата звернення: (Дата звернення: 10.05.2025р)