КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (30), 2025
Спеціальний випуск ISSN 2663 – 4023
Special issue
DOI 10.28925/2663-4023.2025.30.997
УДК 004.94:004.722.5

Соколов Володимир Юрійович
кандидат технічних наук, доцент
доцент кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка
Київський столичний університет імені Бориса Грінченка, м. Київ, Україна
ORCID 0000-0002-9349-7946
v.sokolov@kubg.edu.ua

Новицький Антон Анатолійович
магістр кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка
Київський столичний університет імені Бориса Грінченка, м. Київ, Україна
ORCID 0009-0009-7766-1441
aanovytskyi.fitm24m@kubg.edu.ua

Бодненко Дмитро Миколайович
кандидат педагогічних наук, доцент
доцент кафедри математики і фізики
Київський столичний університет імені Бориса Грінченка, м. Київ, Україна
ORCID 0000-0001-9303-6587
d.bodnenko@kubg.edu.ua

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ КОНФЛІКТНОЇ ВЗАЄМОДІЇ BLE-ПАКЕТІВ

Анотація. Робота присвячена огляду програмно-апаратного методу імітаційного моделювання BLE-пакетів у перенасиченому середовищі з наявністю завадових сигналів. Метод поєднує Android-застосунок, зовнішні BLE-антени на базі ESP32-S3 та nRF52840, а також SDR-приймач HackRF One, що дозволяє формувати й досліджувати конфліктну взаємодію інформативних і завадових радіосигналів у діапазоні 2,4 ГГц. Об'єктом дослідження є конфліктна взаємодія радіосигналів у BLE-каналі. Предметом дослідження є метод моделювання BLE-пакетів із регульованими параметрами та методика експериментального дослідження впливу генератора завад на якість приймання BLE пакетів. Розроблено джерело BLE-реклами з параметрами, такими як канал, інтервал, довжина корисного навантаження та потужність передавання. Також створено оркестратор за допомогою засобів мови програмування Python, який забезпечує роботу множини передавачів nRF52840 з необхідними попередньо заданим параметрами, дані з яких збираються з двох приймачів – зовнішньої BLE-антени підключеною до Android-пристрою та аналізатора пакетів SmartRF. Проведено серію фізичних експериментів без завад і за наявності широкосмугового завадового сигналу. Продемонстровано, що для Android-приймача частка правильно прийнятих кадрів для коротких рекламних пакетів (2–10 байт) без завад становить у середньому 68–78 %, тоді як для SmartRF вона досягає 80–90 %. За введення завадового сигналу спостерігається суттєве (на порядок) зниження ефективної пропускної спроможності каналу та різке зростання кількості «битих» пакетів, особливо для довгих BLE-кадрів. Побудовано теплові карти розподілу кількості отриманих пакетів для nRF і SmartRF. При проведені експерименту проводилось незалежне спостереження за допомогою аналізатора спектру, побудованого на базі SDR HackRF One. Отримані результати можуть бути використані для оцінки завадостійкості BLE-систем та проєктування радіоканалу для IoT-мереж.

Ключові слова: Bluetooth Low Energy, імітаційне моделювання, конфліктна взаємодія сигналів, ESP32, HackRF One, nRF52840, цілісність, доступність, генератор шуму, завада, протидія, завадостійкість радіоканалу.

ВСТУП

Сучасні загальнодоступні системи безпроводового зв’язку широко використовують технологію Bluetooth Low Energy для обміну короткими службовими повідомленнями між малопотужними пристроями. Робота таких систем відбувається в неліцензованому діапазоні 2,4 ГГц, в якому одночасно працює Wi-Fi, безпроводові гарнітури, периферія й іншими радіоінтерфейси. Все це обумовлює підвищений рівень внутрішньоканальної інтерференції та чутливість до завад [1].

Стандартні інструменти тестування BLE-пристроїв переважно орієнтовані на вимірювання енергоспоживання, базових параметрів радіоканалу або функціональну перевірку стеку протоколів і не дають змоги гнучко керувати параметрами рекламних пакетів та відтворювати реалістичні завадові сценарії [2, 3]. Це ускладнює оцінку стійкості до перешкод, особливо у випадку конфігурацій з багатьма пристроями із різними інтервалами та довжинами кадрів. У зв’язку з цим актуальною є задача розробки програмно-апаратного методу моделювання конфліктної взаємодії інформативних та завадових BLE-сигналів, який дозволяє досліджувати втрати пакетів і зміну ефективної пропускної спроможності каналу в контрольованих експериментальних умова.

Постановка проблеми. Об’єктом дослідження є конфліктна взаємодія інформативних і завадових радіосигналів у BLE-каналі. Предметом дослідження є програмно-апаратний метод моделювання BLE-пакетів із регульованими параметрами та методика експериментального дослідження впливу BLE-генератора завад на якість приймання рекламних пакетів. Метою роботи є розробка та експериментальна перевірка методу моделювання BLE-пакетів, який дозволяє:

- формувати множину інформативних BLE-джерел з різними інтервалами і довжинами кадрів;
- моделювати завадові BLE-сигнали в тому самому частотному діапазоні;
- порівнювати роботу різних приймачів (Android-застосунок з зовнішньою антеною та апаратний аналізатор SmartRF) за показниками частки правильно прийнятих, «битих» та втрачених пакетів.

Аналіз останніх досліджень і публікацій. У роботах, які присвячені дослідженню стійкості BLE до активних завад та атак типу глушіння, показують, що навіть при відносно невисоких потужностях завадового сигналу можливе суттєве погіршення якості обслуговування BLE-мережі [4]. Окремі праці зосереджуються на моделюванні атак «людина посередині» на BLE-та інші енергоощадні безпроводові протоколи за допомогою SDR-засобів [5], що підтверджує доцільність використання програмно-визначеного радіо і в задачах дослідження завадостійкості. Разом з тим значна кількість публікацій присвячена застосуванню BLE-маячків для внутрішнього позиціонування за рівнем RSSI [6], де рекламні пакети розглядаються як джерело просторової інформації. У таких роботах питання конфліктної взаємодії рекламних кадрів і завадових сигналів, а також статистики помилок і втрат пакетів зазвичай розглядаються побіжно.

Для аналізу колізій пакетів BLE було розроблено кілька моделей та інструментів моделювання. Спеціальні симулятори та бібліотеки (наприклад, MATLAB Simulink, OpenModelica) дозволяють детально моделювати зв'язок BLE, включаючи структуру пакетів, час та сценарії перешкод [7–11]. Швидкі методи моделювання використовують циклічну природу протоколів BLE для ефективного прогнозування подій колізій, зменшуючи час обчислень [8, 11]. Ці інструменти можуть моделювати як рекламні, так і підключені режими, а також атаки глушіння та перешкод [9, 10].

Збільшення кількості вузлів BLE або зменшення рекламного інтервалу значно підвищує ймовірність колізій пакетів, що призводить до вищого споживання енергії та зниження пропускної здатності [11–13]. Довжина пакета, інтервал з'єднання та алгоритми вибору каналу безпосередньо впливають на частоту колізій та надійність мережі [14–16]. Моделювання показує, що зовнішні перешкоди (наприклад, від Wi-Fi або атак глушіння) можуть погіршити продуктивність BLE, але адаптивні алгоритми та перемикання протоколів можуть пом'якшити ці наслідки [9, 17, 18].

Таким чином, недостатньо дослідженим залишається питання побудови практичного стенду, який дозволяє цілеспрямовано варіювати параметри BLE-пакетів, створювати керовані завадові сценарії та проводити кількісний аналіз розподілу помилок і втрат у різних конфігураціях радіоканалу.

МЕТОДИКА ДОСЛІДЖЕННЯ

Ця стаття присвячена розробці та експериментальному дослідженню методу моделювання Bluetooth з низьким енергоспоживанням ‘Bluetooth Low Energy’ (BLE) пакетів і конфліктної взаємодії інформативних та завадових радіосигналів у діапазоні 2,4 ГГц. У роботі побудовано програмно-апаратний стенд на основі Android-застосунку, зовнішньої BLE-антени nRF52840 та USB-донглу SmartRF, а також проведено серію експериментів із впливу BLE-генератора завад на якість приймання пакетів. Об’єктом дослідження є безпроводовий BLE-радіоканал під впливом електро-магнітних завад. Для проведення дослідження було розроблено Android-застосунок для сканування BLE-пакетів, проведено експериментальні дослідження поведінки BLE-пакетів у сценарії з завадами і без завад засобами створеного Android-застосунку з зовнішньою антеною на nRF5284 і Python-оркестратором, створеного для керування BLE-маячками реклами з параметрами. Методика включає генерацію керованих потоків BLE-реклами з варіюванням інтервалу передавання, довжини корисного навантаження та потужності сигналу в режимах без завади та за наявності RF-генератора шуму, синхронний збір даних двома незалежними приймачами, зіставлення прийнятих кадрів за номером і вмістом корисного навантаження, а також аналіз частки правильних, пошкоджених і втрачених пакетів та зміни ефективної пропускної спроможності каналу для різних сценаріїв конфліктної взаємодії BLE-пакетів.

Для кожної серії експерименту баланс пакетів для одного рекламного пристрою описується співвідношенням:

$$N_{\text{sent}} = N_{ok} + N_{err} + N_{lost}, \quad (1)$$

де N_{sent} – загальна кількість переданих пакетів; N_{ok} – кількість правильно прийнятих пакетів; N_{err} – кількість прийнятих пошкоджених пакетів; N_{lost} – кількість втрачених пакетів.

Вибір апаратного та програмного забезпечення

Для формування інформативних BLE-сигналів використано апаратні засоби на базі мікроконтролерів nRF52840, який застосовуються в якості зовнішніх BLE-антен для Android-застосунку, що дозволяє приймати рекламні пакети, а також використано як керований генератор BLE-кадрів з можливістю тонкого налаштування параметрів реклами.

Приймання корисних BLE-пакетів здійснювалося двома незалежними засобами:

- зовнішньою BLE-антеною на базі nRF52840, підключеною до Android-смартфона (USB-інтерфейс) із спеціалізованим застосунком для логування рекламних кадрів;
- аналізатором BLE-пакетів SmartRF, який передає захоплені кадри на ПК для подальшої обробки.

Також використано SDR HackRF One для демонстрації стану ефіру в діапазоні 2.4 ГГц з завадами і без завад для оцінки відношення SNR.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Схема експерименту

Для нашого експерименту було побудовано систему із застосуванням BLE-пристроїв, кожен з яких за командою від оркестратора відправляє в ефір на попередньо налаштованому каналі рекламні BLE-пакети з певними налаштуваннями. Всі параметри рекламних пакетів, час запуску і інтервал задаються за допомогою Python-оркестратора.

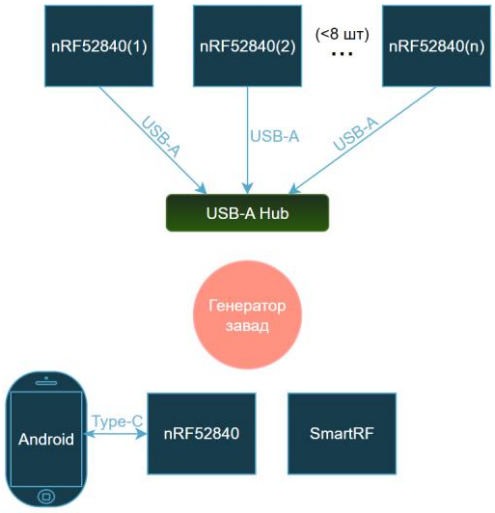


Рис. 1. Схема експерименту

Передавачі розташовуються на відстані 60 см один від одного, приймачі, один з яких Android-застосунок і підключений до нього за допомогою кабелю Type-C – Type-C зовнішньої антени nRF52840, другий з яких SmartRF, розташовуються на відстані 4 метри від передавачів. Також, між приймачами і передавачами розташовано генератор завад.

Точність та фактори навколишнього середовища

Точність результатів даного експерименту визначається часткою правильно прийнятих, пошкоджених та втрачених BLE-пакетів, які вимірювалися в реальних радіочастотних умовах без ізоляції від навколишнього середовища. Експериментальний стенд працює в «бойовому режимі», що означає присутність рекламних пакетів та робочого трафіку від інших BLE-пристроїв, Wi-Fi та іншої периферії. Додатковий внесок у зниження точності для Android-приймача дають обмеження пропускної здатності

інтерфейсу між зовнішньою антеною та смартфоном, які проявляються при надходженні значного обсягу пакетів як від стенду, так і від сторонніх пристроїв.

Підвищення пропускної здатності Android-застосунку, а також оптимізація передачі пакетів з nRF52840 до нього теоретично може призвести до підвищення точності – це вимагає додаткового дослідження.

Розміщення та щільність пристроїв генерації реклами

В експериментальному стенді як маячки використовуються п’ять плат розробників TSTAR MICRO nRF52840, під кожну з яких завантажено однакову прошивку з можливістю налаштування параметрів реклами. Передавачі розташовувались у один ряд на столі на відстані близько 60 см один від одного, що забезпечує достатньо велику щільність джерел сигналу та підвищує ймовірність конфліктної взаємодії їхніх BLE-пакетів на спільному рекламному каналі. Два незалежні приймачі: Android-застосунок із зовнішньою антеною nRF52840 і апаратний аналізатор SmartRF, які розміщувались фронтально до лінії передавачів на відстані приблизно 4 м, а у випадку експериментів із генератором шуму, він встановлювався посередині між передавачами та приймачами на відстані 2 м від кожного. Така схема розміщення та щільність маячків дозволяють моделювати багатопередавацьке середовище з підвищеним рівнем колізій і завад, характерне для реального радіоефіру діапазону 2,4 ГГц.

Джерело реклами TSTAR MICRO nRF52840

Як джерело параметризованої BLE-реклами використано плату TSTAR MICRO nRF52840 з прошивкою на основі Arduino, яка працює напряму з блоком NRF_RADIO без використання повного BLE-стеку. У прошивці введено структуру конфігурації, що задає параметри передавання, яка наведена у лістингу 1.

Лістинг 1. Структура конфігурації параметрів передавача

```
struct TxConfig {
    uint32_t num_packets;
    uint8_t  channel;
    uint8_t  payload_len;
    uint32_t interval_ms;
    int8_t   tx_power_dbm;
};

static TxConfig g_cfg = {   100,  39,    20,    10,    0};
```

Налаштування кожного передавача виконується з боку оркестратора через послідовний інтерфейс за двома текстовими командами, які наведено в лістингу 2.

Лістинг 2. Текстові команди для керування передавачем

```
SET C=<N> CH=<37/38/39> PL=<len> I=<ms> P=<dBm>
GO
```

Команда SET розбирається у функції обробки рядка, яка наведена в лістингу 3, й оновлює поля структури g_cfg, яке наведено в лістингу 1.

Лістинг 3. Функція обробки команди SET

```
static void handle_set_command(char *args)
{
    while (*args == ' ') args++;
    char *token = strtok(args, " ");
    while (token != NULL) {
        if (strncmp(token, "C=", 2) == 0) {
            g_cfg.num_packets = (uint32_t)strtoul(token + 2, NULL,10);
        } else if (strncmp(token, "CH=", 3) == 0) {
            g_cfg.channel = (uint8_t)strtoul(token + 3, NULL, 10);
        } else if (strncmp(token, "PL=", 3) == 0) {
            g_cfg.payload_len = (uint8_t)strtoul(token + 3, NULL, 10);
        } else if (strncmp(token, "I=", 2) == 0) {
            g_cfg.interval_ms = (uint32_t)strtoul(token + 2, NULL,10);
        } else if (strncmp(token, "P=", 2) == 0) {
            g_cfg.tx_power_dbm = (int8_t)strtol(token + 2, NULL, 10);
        }
        token = strtok(NULL, " ");
    }
    apply_config();
    Serial.println("OK");
}
```

Функція apply_config виконує прив’язку параметрів до апаратного радіомодуля: встановлює номер каналу, потужність передавання, формат кадру (довжину поля даних) та формує рекламну адресу на основі унікальних регістрів NRF_FICR->DEVICEADDR[x]. Після надходження команди GO прошивка обнуляє лічильник кадрів і переходить до циклу передавання, алгоритм якого наведено в лістингу 4.

Лістинг 4. Цикл передавання пакетів на рекламному пристрої

```
if (g_start_flag) {
        g_start_flag = false; int8_t payload[32]; uint32_t t_next = micros();
        for (uint32_t i = 0; i < g_cfg.num_packets; ++i) {
            t_next += (uint32_t)g_cfg.interval_ms * 1000UL;
            if (g_cfg.payload_len > 0) {
                const uint16_t cnt = g_packet_counter++;
                payload[0] = static_cast<uint8_t>(cnt & 0xFF);
                if (g_cfg.payload_len > 1) {
                    payload[1] = static_cast<uint8_t>((cnt >> 8) & 0xFF);
                }
                if (g_cfg.payload_len > 2) {
                    memset(payload + 2, 0xAA, g_cfg.payload_len - 2);
                }
            }
            radio_send_packet(payload);
            while ((int32_t)(micros() - t_next) < 0) {
            }
        }Serial.println("DONE");}
```

Отже за допомогою вищенаведеного прикладу реалізації пристрою для генерації BLE-реклами, TSTAR MICRO nRF52840 виступає як керований генератор BLE-рекламних кадрів з можливістю для оркестратора задавати кількість пакетів, канал, довжину корисного навантаження, інтервал між кадрами та рівень потужності, а також вшивати в payload лічильник кадру для подальшого аналізу втрат і помилок.

Оркестратор експерименту

Для керування декількома джерелами реклами TSTAR MICRO nRF52840 використано окремий оркестратор, який реалізований засобами мови програмування Python із графічним інтерфейсом (див. рис. 2) на базі *tkinter*.

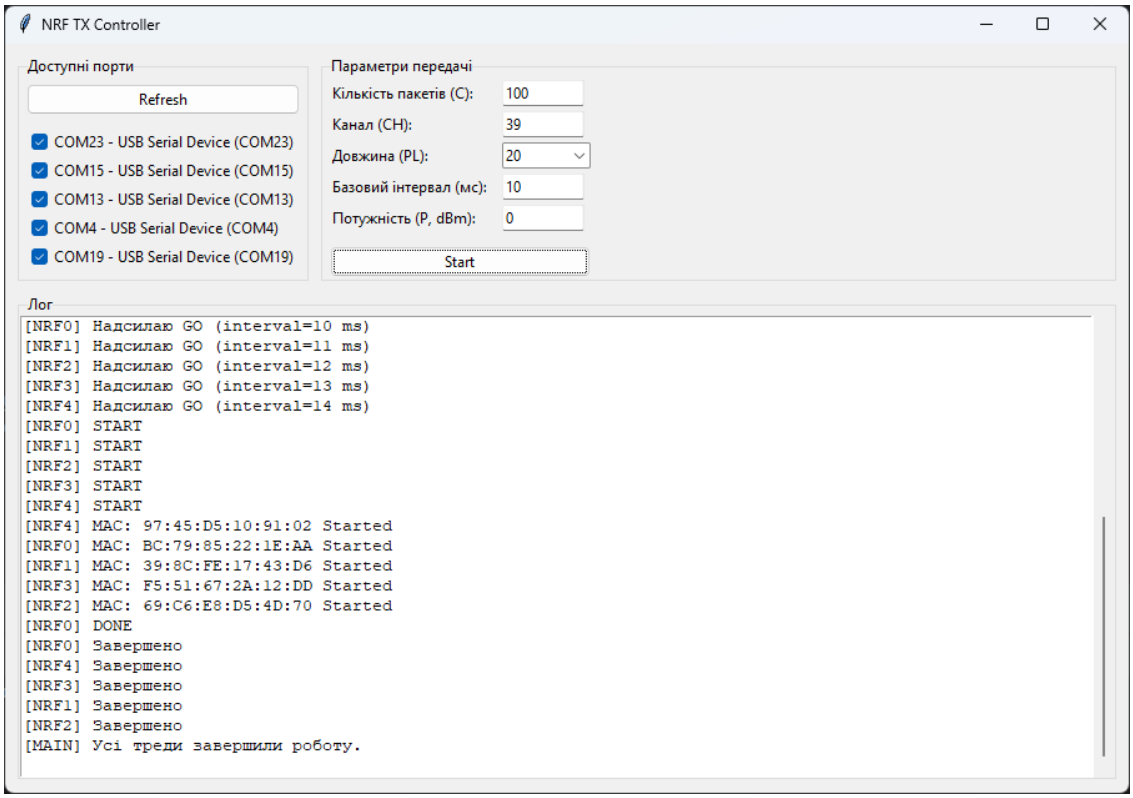


Рис. 2. Загальний вид графічного інтерфейсу оркестратора

Оркестратор працює у багатопотоковому режимі, де для кожного вибраного COM-порту створюється окремий робочий потік NRFWorker, якому передається структура параметрів наведена в лістингу 5.

Лістинг 5. Структура параметрів Python-оркестратора

```
class NRFWorker(threading.Thread):
    def run(self):
        interval_ms = int(self.cfg.base_interval_ms * (1.0 + 0.1 *
self.cfg.index))
        ser = serial.Serial(self.cfg.port, baudrate=BAUDRATE, timeout=1)
        cmd = (
            f "SET"
            f" C={self.cfg.num_packets}"
            f" CH={self.cfg.channel}"
            f" PL={self.cfg.payload_len}"
            f" I={interval_ms}"
            f" P={self.cfg.tx_power_dbm}\n"
        )
        self.log(f"Надсилаю конфіг: {cmd.strip()}")
        ser.write(cmd.encode("ascii"))
        ser.flush()
```

```
        time.sleep(0.1)
        ack = ser.readline().decode(errors="ignore").strip()
        if ack:
            self.log(f"Відповідь: {ack}")
        self.log("Очікую глобальний START")
        self.start_event.wait()
        self.log(f"Надсилаю GO (interval={interval_ms} ms)")
        ser.write(b"GO\n")
        ser.flush()
        while True:
            line = ser.readline().decode(errors="ignore").strip()
            if not line:
                break
            self.log(line)
            if line == "DONE":
                break
        ser.close()
        self.log("Завершено")
```

Графічний інтерфейс оркестратора відображає список доступних портів, дозволяє позначити потрібні пристрої та задати загальні параметри передачі. Після натискання кнопки *Start* програма збирає вибрані порти, створює для кожного з них DeviceConfig і запускає відповідні потоки NRFWorker, використовуючи спільну подію start_event для старту всіх передавачів.

Всі повідомлення від потоків збираються та відображаються у вікні журналу, що дозволяє в реальному часі контролювати надсилання команд SET або GO і завершення роботи кожного рекламного пристрою. Таким чином, оркестратор забезпечує синхронний запуск серій BLE-пакетів на кількох nRF52840 з єдиного інтерфейсу та гарантує відтворюваність умов експерименту.

Результати збору даних та уніфікації експериментальних даних

Після виконання кожної серії експериментів дані одночасно реєструються двома незалежними приймачами: Android-приймачем із зовнішньою BLE-антеною на основі nRF52840, в якому формат файлів логів ‘.ndjson’ та апаратним сніфером SmartRF Packet Sniffer, в якому формат файлів логів ‘.psd’. Для подальшого аналізу ці різні за форматом журнали приводяться до єдиного вигляду за допомогою Python-скриптів.

На першому етапі обробки скрипт рекурсивно проходить папку зберігання файлів логів приймачів, знаходить пари файлів для кожного режиму експерименту від кожного з приймачів з однаковими параметрами кількості пакетів, каналу, інтервалу, довжини payload і потужності та перетворює їх у табличне представлення. Для Android-файлів кожен рядок містить корисне навантаження пакета у вигляді hex-рядка під назвою ‘payload_hex’ та MAC-адресу під назвою ‘mac’. Для SmartRF-файлів кожен рядок містить корисне навантаження в полі ‘col3’ та MAC-адресу в полі ‘col2’. Витяг корисного навантаження й попередню фільтрацію пакетів за MAC-адресами реалізовано, зокрема, такою функцією, що наведена в Лістингу 6.

Лістинг 6. Функція форматування отриманих пакетів з Android-приймача

```
ALLOWED_MAC = ["39:8C:FE:17:43:D6",..., "69:C6:E8:D5:4D:70"]
def extract_from_ndjson(path: str):
    base_out_path = os.path.splitext(path)[0] + ".txt"
    file_name = os.path.basename(base_out_path)
    ndjson_dir = os.path.dirname(path)
```

```
    dir_name = os.path.basename(ndjson_dir)
    name_of_experiment = os.path.basename(os.path.dirname(ndjson_dir))
       out_dir=os.path.join(os.curdir,"results_postprocessed",name_of_experiment,dir
_name)
    os.makedirs(out_dir, exist_ok=True)
    out_path = os.path.join(out_dir, file_name)
    with open(path, "r", encoding="utf-8") as src, \
         open(out_path, "w", encoding="utf-8") as dst:
        for line_no, line in enumerate(src, 1):
            line = line.strip()
            if not line:
                continue
            try:
                obj = json.loads(line)
            except json.JSONDecodeError:
                dst.write(f"# line {line_no} (invalid JSON): {line}\n")
                continue
            if 'mac' not in obj or obj['mac'] not in ALLOWED_MAC:
                continue
            if isinstance(obj, dict):
                chosen = None
                for key in POSSIBLE_PACKET_KEYS:
                    if key in obj:
                        chosen = obj[key]
                        break
                if chosen is not None:
                    if isinstance(chosen, (bytes, bytearray)):
                        dst.write(binascii.hexlify(chosen).decode("ascii") + "\n")
                    else:
                        dst.write(str(chosen) + "\n")
                    continue
            dst.write(json.dumps(obj, ensure_ascii=False) + "\n")
    print(f"[NDJSON] {path} -> {out_path}")
```

Аналогічно для SmartRF файлів, з яких зчитуються «сирі» кадри packetRaw, з них виділяються MAC-адреса та ADV-payload, а на вихід записуються рядки у форматі «packetNumber;MAC;payload_hex», код якого наведено в Лістингу 7. Це дозволяє мати для обох приймачів уніфіковані текстові файли, з яких далі легко формуються pandas фрейми.

Лістинг 7. Функція форматування отриманих пакетів з SmartRF-приймача

```
def  extract_from_psd(path:str,header_size:int=PSD_HEADER_SIZE,record_size:  int  =
PSD_RECORD_SIZE):
    base_out_path = os.path.splitext(path)[0] + ".txt"
    file_name = os.path.basename(base_out_path)
    ndjson_dir = os.path.dirname(path)
    dir_name = os.path.basename(ndjson_dir)
    name_of_experiment = os.path.basename(
        os.path.dirname(ndjson_dir)
    )
    out_dir = os.path.join(
        os.curdir,
        "results_postprocessed",
        name_of_experiment,
        dir_name,
    )
    os.makedirs(out_dir, exist_ok=True)
    out_path = os.path.join(out_dir, file_name)
    with open(path, "rb") as f:
        with open(out_path, "w", encoding="utf-8") as out:
            while True:
```

```
chunk = f.read(PSD_RECORD_SIZE)
if not chunk:
    break
if len(chunk) != PSD_RECORD_SIZE:
    print(f"Попередження: урізаний запис "
        f"({len(chunk)} байт замість {PSD_RECORD_SIZE}).")
    break
(
    packetInfo,
    packetNumber,
    timeStamp,
    payloadLength,
    packetLen,
    packetRaw,
) = struct.unpack(PSD_STRUCT_FMT, chunk)
mac = extract_mac_from_packet(packetRaw, packetLen)
if mac not in ALLOWED_MAC:
    continue
if packetLen < 2:
    print(f"Пропускаю пакет {packetNumber}: packetLen={packetLen} < 2")
    continue
payload = extract_advdata_from_packet_data_hex(packetRaw.hex())
line = f"{packetNumber};{mac};{payload}\n"
out.write(line)
```

На другому етапі для кожного пакета відновлюється номер кадру. У BLE-пакетах перші два байти корисного навантаження є лічильником у діапазоні 0–999, закодованим у форматі little-endian. Отримане значення зберігається в колонці packet_no_dec, після чого датафрейми для Android-приймача і SmartRF сортуються за цим номером. Для оцінки правильності пакетів вводиться ознака ‘is_clean’, яка визначається за простими правилами перевірки payload:

- довжина дорівнює чотирьом символам і номер менший 1000;
- довжина більше ніж чотири символи, то всі байти після перших двох мають збігатися з заповнювачем 0xAA і номер менший за 1000.

В результаті для кожної MAC-адреси й для кожного з приймачів окремо автоматично обчислюються загальна кількість прийнятих пакетів, кількість правильних та неправильних пакетів. Додатково формуються множини пар ‘номер пакета-payload’ для Android і SmartRF, на основі яких підраховуються повні співпадіння і часткові співпадіння, де однаковий номер, але різні заповнювачі.

Для кількісної оцінки впливу параметрів реклами на роботу радіоканалу, окрім відносної кількості правильно прийнятих кадрів, доцільно ввести показник ефективної пропускної здатності:

$$R_{\text{eff}} = \left(\frac{8\,L_{\text{payload}}}{T_{\text{int}}}\right) p_{\text{ok}}, \tag{2}$$

де R_{eff} – ефективна пропускна здатність каналу, біт/с; L_{payload} – довжина корисного навантаження, байт; T_{int} – інтервал між рекламними пакетами, с; p_{ok} – частка правильно прийнятих пакетів.

На основі цих розрахунків для кожного режиму сформовані табл. 1 по всіх MAC-адресах.

Таблиця 1

Кількість отриманих правильних пакетів за допомогою nRF і SmartRF

Кількість отриманих даних, байт	Інтервал, мс	nRF, %	SmartRF, %
30	10	63,3±24,3	79,3±10,2
	20	62,3±19,4	41,8±3,2
	40	1,9±0,2	50,0±34,0
20	10	66,3±15,2	66,3±15,2
	20	64,1±29,1	64,1±29,1
	40	2,1±0,9	2,1±0,9
10	10	68,3±8,0	78,6±11,2
	20	62,5±13,9	85,8±2,2
	40	19,0±7,4	59,1±16,0
2	10	78,8±8,0	86,5±6,1
	20	66,8±11,7	84,8±79,0
	40	32,6±7,3	91,4±38,0

Паралельно проводилася оцінка стану ефіру на частоті 2,480 ГГц, що відповідає 39 каналу, на якій працюють рекламні пристрої в експерименті. За спектрограмою, отриманою за допомогою SDR HackRF One (див. рис. 3), видно, що в ефірі присутня значна кількість сторонніх BLE-маячків і іншого трафіку в діапазоні 2,4 ГГц, тобто система працює не в ізольованій лабораторній обстановці, а в «живому» перевантаженому ефірі. Це безпосередньо впливає на втрати пакетів і на відмінності в роботі Android-приймача та SmartRF.

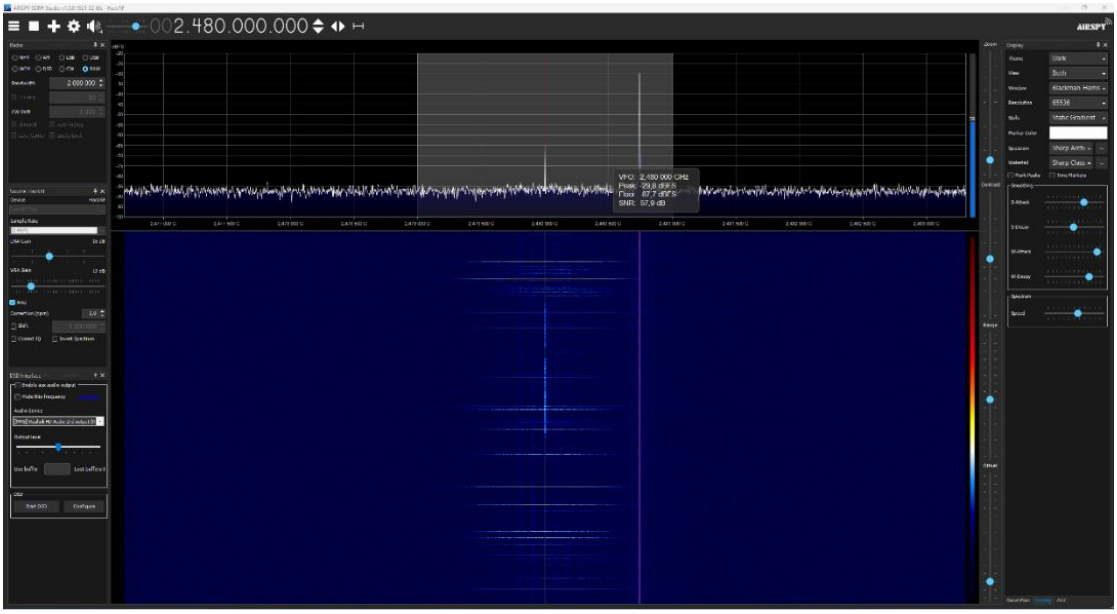
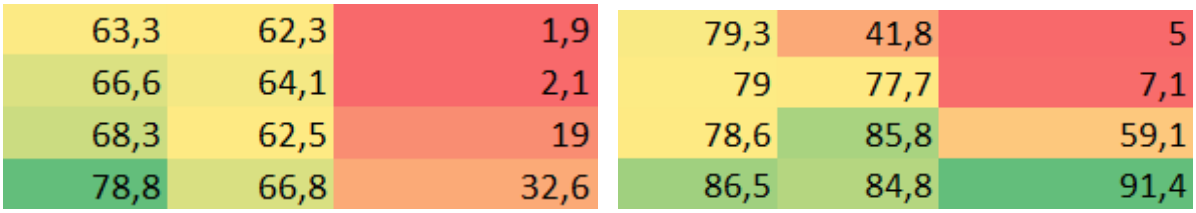


Рис. 3. Аналіз ефіру 39 каналу BLE за допомогою HackRF One

Аналіз даних для експерименту без завад виконано на основі табл. 1 та відповідних теплових карт (див. рис.4).



а б

Рис. 4. Теплова карта розподілу кількості отриманих пакетів для nRF (а) і SmartRF (б)

Для Android-приймача з зовнішньою антеною nRF в режимах з малими інтервалами реклами (від 10 до 20 мс) й довгими кадрами (від 20 до 30 байт) частка правильно прийнятих пакетів зазвичай становить близько 60–66%, а для коротких кадрів (від 2 до 10 байт) зростає до 68–78 %. При збільшенні інтервалу до 40 мс частка правильно прийнятих кадрів різко падає до одиниць відсотків, що добре видно на теплових картах для nRF-приймача, що зображено на тепловій карті на рис. 4а і табл. 1.

Для SmartRF у тих самих режимах частка правильно декодованих кадрів для довгих payload коливається в межах від 41 до 80%, а для коротких та середніх (від 2 до 20 байт) досягає від 60 до 90 %, що зображено на тепловій карті на рис. 4б і табл. 1. Тобто апаратний сніффер загалом приймає більше коректних кадрів, особливо при довгих payload.

Згідно табл. 2 кількість «битих» пакетів на nRF-приймачі незначна: у більшості режимів це від 1 до 3 пакетів на 1000, з локальним зростанням до 11 пакетів при інтервалі 40 мс і довжині 2 байти.

Для SmartRF кількість пакетів із помилками в payload суттєво більша, а саме, до кількох десятків на 1000 переданих в деяких режимах, що пояснюється його спроможністю декодувати значно більшу кількість гранично ослаблених та зашумлених кадрів, які Android-приймач взагалі не реєструє.

Таблиця 2

Кількість отриманих неправильних пакетів за допомогою nRF і SmartRF

Кількість отриманих даних, байт	Інтервал, мс	nRF, %	SmartRF, %
30	10	2,5	42,5
	20	0,8	32,8
	40	0	13,0
20	10	1,8	38,8
	20	1,0	52,0
	40	0,3	6,0
10	10	0,8	28,8
	20	1,0	13,8
	40	0,3	23,0
2	10	2,0	0,3
	20	0,3	2,5
	40	11,0	1,5

Узгодження прийнятих пакетів між двома приймачами (див. табл. 3) показує, що для режимів з короткими інтервалами й невеликими payload кількість повних співпадінь досягає приблизно 600 пакетів, а часткових не перевищує декількох десятків.

Таблиця 3

Кількість співпадінь отриманих пакетів за допомогою nRF52840 і SmartRF

Кількість отриманих даних, байт	Інтервал, мс	Повне співпадіння, ‰	Часткове співпадіння, ‰
30	10	505	6
	20	371	11
	40	12	3
20	10	573	4
	20	352	1
	40	17	0
10	10	611	8
	20	496	24
	40	134	2
2	10	609	0
	20	530	1
	40	155	0

Це свідчить, що в більшості випадків обидва приймачі або не бачать пакет зовсім, або декодують його однаково, а систематичних розбіжностей у декодуванні payload не спостерігається.

Отримані залежності також демонструють вплив навколишнього середовища та «засміченості» ефіру: збільшення інтервалу реклами до 40 мс в умовах великої кількості сторонніх BLE-передавачів призводить до значного падіння частки правильно прийнятих кадрів на Android-пристрої. Це пов’язано з тим, що між власними кадрами nRF-маячків приймач активно сканує й передає на Android-застосунок пакети від інших пристроїв; у пікові моменти частина кадрів губиться вже на шляху від донгла до застосунку по USB. Для SmartRF, який безпосередньо підключений до ПК з більшими ресурсами і пропускною спроможністю інтерфейсу, такої різкої деградації не спостерігається.

В експерименті з завадою методика обробки даних залишається тією самою, а результати аналізу наведено у підсумковій табл. 4.

Таблиця 4

Результати роботи nRF52840 і SmartRF із завадою при інтервалі передачі 10 і 20 мс

Кількість пакетів	Пристрій	Інтервал передачі, мс							
		10	20	10	20	10	20	10	20
		Передавач 1		Передавач 2		Передавач 3		Передавач 4	
Відправлено	nRF52840	1000							
Отримано	nRF52840	4	1	2	2	2	1	1	—
без помилки		4	1	2	2	2	1	1	—
з помилкою		0	0	0	0	0	0	0	—
Отримано	SmartRF	0	0	0	0	0	0	0	—
без помилки		0	0	0	0	0	0	0	—
з помилкою		0	0	0	0	0	0	0	—

Попередній спектральний аналіз за допомогою HackRF One показав, що при увімкненому генераторі сигнал завади домінує в діапазоні, а співвідношення сигнал/шум (SNR) на робочій частоті суттєво знижується (порівняно без завади 57,9 дБмВт – див. рис. 3, при роботі генератора SNR близько 11,4 дБмВт – див. рис. 5).

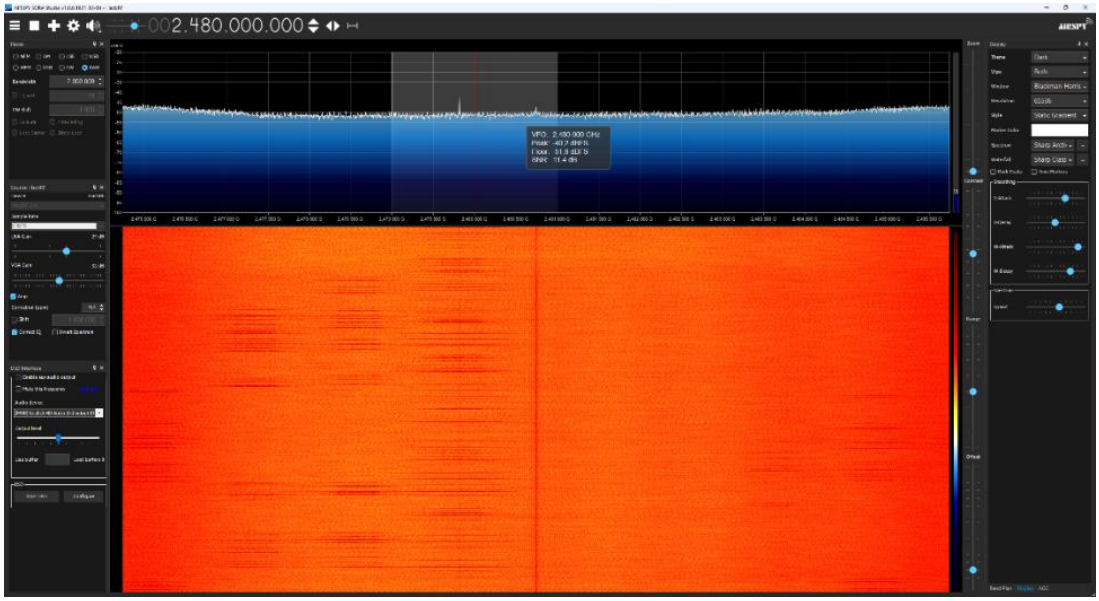


Рис. 4. Демонстрація роботи генератора завад за допомогою SDR HackRF One

В даному випадку ми також можемо застосувати стандартну формулу для визначення відношення сигналу до шуму:

$$SNR^{\mathrm{dB}} = P_{\mathrm{sig}}^{\mathrm{dBm}} - P_{\mathrm{noise}}^{\mathrm{dBm}} \quad (3)$$

де $P_{\mathrm{sig}}^{\mathrm{dBm}}$ – потужність корисного сигналу, $P_{\mathrm{noise}}^{\mathrm{dBm}}$ – потужність шуму/завади.

За таких умов навіть при максимальній доступній потужності передавачів nRF52840 (8 дБмВт) радіоканал практично перестає бути придатним для передавання інформаційних BLE-пакетів.

Згідно табл. 4, для інтервалів реклами від 10 і 20 мс та довжини payload 2 байти з 1000 переданих кадрів кожним із чотирьох передавачів Android-приймач реєструє лише до 4 правильно декодованих пакетів, тоді як SmartRF у тих самих режимах не фіксує жодного пакета. Це підтверджує, що за наявності потужної широкосмугової завади на робочому каналі рекламний трафік практично повністю заглушується. Водночас, навіть в цих умовах Android-застосунок інколи отримує одиничні кадри, що формально демонструє кращий результат порівняно з донглом SmartRF, хоча абсолютна частка прийнятих пакетів залишається меншою за відсоток.

Узагальнюючи, розроблена схема обробки й уніфікації даних дозволяє з «сирих» JSON-логів Android-приймача та PSD-файлів SmartRF отримати єдиний узгоджений датасет з відновленими номерами пакетів, класифікацією «правильний/поламаний» та статистикою повних і часткових співпадінь між приймачами.

ОБГОВОРЕННЯ ТА РЕКОМЕНДАЦІЇ ДО ВПРОВАДЖЕННЯ

Запропонований у роботі метод і апаратно-програмний стенд дають змогу відносно дешевими засобами відтворювати реальні сценарії роботи BLE у перевантаженому ефірі та оцінювати, як параметри реклами впливають на частку правильно прийнятих, «битих» і втрачених пакетів для різних приймачів [19]. Експерименти показали, що в умовах реального радіоефіру, де одночасно працює значна кількість сторонніх BLE [4], ZigBee [20] і Wi-Fi пристроїв [21–23], найбільш стійкими до втрат є режими з короткими інтервалами реклами (від 10 до 20 мс) і малими або середніми payload (від 2 до 20 байт), але збільшення інтервалу до 40 мс на фоні стороннього трафіку призводить до різкого падіння кількості правильно прийнятих кадрів на Android-приймачі.

Наведені результати є продовженням роботи із визначення розподілу помилок при передаванні BLE-пакетів [24] і демонструють можливі недоліки при впровадженні BLE-рішень, які можна уникнути за допомогою наведеного експериментального стенду. По-перше, для систем, де важлива стійкість до втрат (телеметрія, службові маячки або інше), доцільно обирати короткі рекламні інтервали та мінімізувати довжину корисного навантаження, особливо в умовах перенасиченого ефіру. По-друге, при масовому зборі статистики краще спиратися на апаратний сніфер на кшталт SmartRF, а Android-застосунок із зовнішньою антеною розглядати як більш обмежений за пропускною здатністю інструмент з даною реалізацією, чутливий до перевантаження USB-каналу і стороннього трафіку - це вимагає додаткового дослідження і вдосконалення Android-застосунку і прошивки зовнішньої антени до нього. По-третє, перед розгортанням систем, для яких важливий захист від завад, доцільно проводити попередній спектральний аналіз ефіру (наприклад, за допомогою SDR HackRF One) і тестувати стійкість до навмисних завад: результати експерименту показують, що потужний широкосмуговий генератор шуму здатний практично повністю заблокувати рекламний канал навіть при максимальній доступній потужності передавачів nRF52840.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

В роботі розроблено та експериментально досліджено програмно-апаратний метод моделювання BLE-пакетів, який дозволяє створювати налаштовувальні джерела реклами на базі nRF52840, Python-оркестратор для керованого запуску передавачів та два незалежні приймачі: Android-застосунок із зовнішньою BLE-антеною й апаратний сніффер SmartRF. Запропонована схема збору й обробки даних дає змогу кількісно оцінювати розподіл помилок і втрат у різних режимах роботи рекламного каналу. Показано, що при оптимальних комбінаціях параметрів більшість кадрів приймаються обома приймачами, тоді як збільшення інтервалу реклами й довжини payload у реальному зашумленому ефірі призводить до різкого погіршення прийому на Android-пристрої. Експерименти з генератором завад підтвердили, що за наявності потужного широкосмугового шуму рекламний канал практично повністю заблокований.

Подальші дослідження доцільно спрямувати у кількох напрямках. По-перше, на вдосконалення Android-застосунку та прошивки антени nRF52840 (оптимізація обробки потоку пакетів, FIFO-черги для передачі з антени до смартфону) з метою зменшення втрат у вузьких місцях. По-друге, на розширення набору сценаріїв: дослідження не лише рекламних, а й з’єднаних режимів BLE, інших шаблонів корисного навантаження, різних типів завад (вузькосмугові, спрямовані на окремі канали тощо). По-третє, на використання створеного стенду як інструмента для оцінки завадостійкості та інформаційної безпеки BLE-рішень у прикладних задачах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hulak, H., Zhyltsov, O., Kyrychok, R., Korshun, N., & Skladannyi, P. (2023). Enterprise Information and Cyber Security. [*Textbook*]. Borys Grinchenko Kyiv Metrop. Univ.
2. Kostiuk, Y., Skladannyi, P., Bebeshko, B., Khorolska, K., Rzaieva, S., & Vorokhob, M. (2025). Information and Communication Systems Security. [*Textbook*]. Borys Grinchenko Kyiv Metrop. Univ.
3. Kostiuk, Y., Skladannyi, P., Hulak, H., Bebeshko, B., Khorolska, K., & Rzaieva, S. (2025). Information Security Systems. [*Textbook*]. Borys Grinchenko Kyiv Metrop. Univ.
4. Sokolov, V., Skladannyi, P., & Astapenya, V. (2023). Bluetooth Low-Energy Beacon Resistance to Jamming Attack. In *2023 IEEE 13th Int. Conf. on Electronics and Information Technologies (ELIT)* (pp. 270–274).
5. TajDini, M., Sokolov, V., & Buriachok, V. (2019). Men-in-the-Middle Attack Simulation on Low Energy Wireless Devices using Software Define Radio. In *8th Int. Conf. on "Mathematics. Information Technologies. Education:" Modern Machine Learning Technologies and Data Science*, vol. 2386 (pp. 287–296).
6. Pakanon, N., Chamchoy, M., & Supanakoon, P. (2020). Study on Accuracy of Trilateration Method for Indoor Positioning with BLE Beacons. In *2020 6th Int. Conf. on Engineering, Applied Sci. and Technol.* https://doi.org/10.1109/ICEAST50382.2020.9165464
7. Ghamari, M., Villeneuve, E., Soltanpur, C., Khangosstar, J., Janko, B., Sherratt, R., & Harwin, W. (2018). Detailed Examination of a Packet Collision Model for Bluetooth Low Energy Advertising Mode. *IEEE Access, 6*, 46066–46073. https://doi.org/10.1109/access.2018.2866323
8. Kindt, P., Park, S., & Chakraborty, S. (2018). Fast Collision Simulation for Cyclic Wireless Protocols. *ArXiv*. https://doi.org/10.48550/arXiv.1807.01645
9. Raj, M., & Achuthan, K. (2022). Transmission Modeling and Attack Simulation of Bluetooth Low Energy. In *2022 2nd Int. Conf. on Interdisciplinary Cyber Physical Systems (ICPS)* (pp. 93–98). https://doi.org/10.1109/icps55917.2022.00025
10. Shavelis, R., & Ozols, K. (2020). Bluetooth Low Energy Wireless Sensor Network Library in MATLAB Simulink. *J. Sens. Actuator Netw., 9, 38*. https://doi.org/10.3390/jsan9030038
11. Shan, G., Lee, B., Shin, S., & Roh, B. (2017). Design and Implementation of Simulator for Analysis of BLE Broadcast Signal Collision. In *2017 Int. Conf. on Information Networking (ICOIN)* (pp. 448–452). https://doi.org/10.1109/icoin.2017.7899533
12. Nikodem, M., & Bawiec, M. (2019). Experimental Evaluation of Advertisement-based Bluetooth Low Energy Communication. *Sensors, 20*. https://doi.org/10.3390/s20010107
13. Nikodem, M., Słabicki, M., & Bawiec, M. (2020). Efficient Communication Scheme for Bluetooth Low Energy in Large Scale Applications. *Sensors, 20*. https://doi.org/10.3390/s20216371
14. Pang, B., Claeys, T., Vankeirsbilck, J., T'Jonck, K., Hallez, H., & Boydens, J. (2023). A Novel Model to Quantify the Impact of Transmission Parameters on the Coexistence Between Bluetooth Low Energy Pairs. *IEEE Internet Things J., 10*, 1733–1745. https://doi.org/10.1109/jiot.2022.3209874
15. Pang, B., T'Jonck, K., Claeys, T., Pissoort, D., Hallez, H., & Boydens, J. (2021). Bluetooth Low Energy Interference Awareness Scheme and Improved Channel Selection Algorithm for Connection Robustness. *Sensors, 21*. https://doi.org/10.3390/s21072257
16. Kim, M. (2017). An Analysis of Packet Collision Probability due to Inter-Piconet Interference in the Bluetooth Low Energy Networks. *J. Inst, Electr. Eng. Korea, 54*, 3–11. https://doi.org/10.5573/ieie.2017.54.8.3
17. Del-Valle-Soto, C., Valdivia, L., Velázquez, R., Del-Puerto-Flores, J., Varela-Aldás, J., & Visconti, P. (2025). Adaptive Jamming Mitigation for Clustered Energy-Efficient LoRa-BLE Hybrid Wireless Sensor Networks. *Sensors, 25*. https://doi.org/10.3390/s25061931
18. Park, W., Eo, S., Joo, C., & Bahk, S. (2024). B-Hop: Time-Domain Adjustment of BLE Frequency Hopping Against Wi-Fi Beacon Interference. *IEEE Internet Things J., 11*, 12853–12863. https://doi.org/10.1109/jiot.2023.3337252
19. Sokolov, V., Skladannyi, P., & Platonenko, A. (2022). Video Channel Suppression Method of Unmanned Aerial Vehicles. In *2022 IEEE 41st International Conference on Electronics and Nanotechnology (ELNANO)* (pp. 473–477). https://doi.org/10.1109/elnano54667.2022.9927105
20. Sokolov, V., Skladannyi, P., & Korshun, N. (2023). ZigBee Network Resistance to Jamming Attacks. In *2023 IEEE Int. Conf. on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo)* (pp. 161–165). https://doi.org/10.1109/ukrmico61577.2023.10380360

21. Sokolov, V., Skladannyi, P., & Mazur, N. (2023). Wi-Fi Repeater Influence on Wireless Access. In *2023 IEEE 5th Int. Conf. on Advanced Information and Communication Technologies (AICT)* (pp. 33–36). https://doi.org/10.1109/aict61584.2023.10452421
22. Sokolov, V., Skladannyi, P., & Platonenko, A. (2023). Jump-Stay Jamming Attack on Wi-Fi Systems. In *2023 IEEE 18th Int. Conf. on Computer Science and Information Technologies (CSIT)* (pp. 1–5). https://doi.org/10.1109/csit61576.2023.10324031
23. Sokolov, V., Skladannyi, P., & Astapenya, V. (2023). Wi-Fi Interference Resistance to Jamming Attack. In *2023 IEEE 5th Int. Conf. on Advanced Information and Communication Technologies (AICT)* (pp. 1–4). https://doi.org/10.1109/aict61584.2023.10452687
24. Novytskyi, A., Sokolov, V., Kriuchkova, L., & Skladannyi, P. (2025). Determining the Error Distribution of BLE Beacons at Antenna Near and Far Fields. In *4th Int. Conf. on Cyber Hygiene & Conflict Management in Global Information Networks (CH&CMiGIN'2025)*, vol. 4024 (pp. 133–143).

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (30), 2025
Спеціальний випуск ISSN 2663 – 4023
Special issue

Volodymyr Y. Sokolov
Ph.D., associate professor
associate professor of the Department of Information and Cyber Security
named after Professor Volodymyr Buriachok
Borys Grinchenko Kyiv Metropolitan University, Kyiv, Ukraine
ORCID 0000-0002-9349-7946
v.sokolov@kubg.edu.ua

Anton A. Novytskyi
master of the Department of Information and Cyber Security
named after Professor Volodymyr Buriachok
Borys Grinchenko Kyiv Metropolitan University, Kyiv, Ukraine
ORCID 0009-0009-7766-1441
aanovytskyi.fitm24m@kubg.edu.ua

Dmytro M. Bodnenko
Ph.D., associate professor
associate professor at the Department of Mathematics and Physics
Borys Grinchenko Kyiv Metropolitan University, Kyiv, Ukraine
ORCID 0000-0001-9303-6587
d.bodnenko@kubg.edu.ua

SIMULATION MODELING OF CONFLICT INTERACTION OF BLE PACKETS

Abstract. The article is devoted to reviewing the software-hardware method of simulating BLE packets in a saturated environment with the presence of interfering signals. The method combines an Android application, external BLE antennas based on ESP32-S3 and nRF52840, as well as the HackRF One SDR receiver, allowing you to form and study the interaction of informative and interfering radio signals in the 2.4 GHz range. The object of the study is the conflict interaction of radio signals in the BLE channel. The subject of the study is a method for modeling BLE packets with adjustable parameters and a methodology for experimentally investigating the impact of an interference generator on the quality of BLE packet reception. A BLE advertising source with parameters such as channel, interval, payload length, and transmission power has been developed. An orchestrator was also created using the Python programming language, which ensures the operation of a set of nRF52840 transmitters with the necessary preset parameters. The data from these transmitters is collected by two receivers: an external BLE antenna connected to an Android device and a SmartRF packet analyzer. A series of physical experiments was conducted without interference and in the presence of a broadband interference signal. It was demonstrated that for the Android receiver, the proportion of correctly received frames for short advertising packets (2–10 bytes) without interference is on average 68–78%, while for SmartRF it reaches 80–90%. When an interference signal is introduced, a significant (by an order of magnitude) decrease in the adequate channel bandwidth is observed, and a sharp increase in the number of “broken” packets, especially for extended BLE frames. Heat maps of the distribution of the number of received packets for nRF and SmartRF were constructed. The experiment was conducted using an independent spectrum analyzer built on the HackRF One SDR. The results can be used to assess the noise immunity of BLE systems and design a radio channel for IoT networks.

Keywords: Bluetooth Low Energy, simulation modeling, conflicting signal interaction, ESP32, HackRF One, nRF52840, integrity, availability, noise generator, interference, countermeasure, radio channel interference immunity.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Hulak, H., Zhyltsov, O., Kyrychok, R., Korshun, N., & Skladannyi, P. (2023). Enterprise Information and Cyber Security. [*Textbook*]. Borys Grinchenko Kyiv Metrop. Univ.
2. Kostiuk, Y., Skladannyi, P., Bebeshko, B., Khorolska, K., Rzaieva, S., & Vorokhob, M. (2025). Information and Communication Systems Security. [*Textbook*]. Borys Grinchenko Kyiv Metrop. Univ.

3. Kostiuk, Y., Skladannyi, P., Hulak, H., Bebeshko, B., Khorolska, K., & Rzaieva, S. (2025). Information Security Systems. [*Textbook*]. Borys Grinchenko Kyiv Metrop. Univ.
4. Sokolov, V., Skladannyi, P., & Astapenya, V. (2023). Bluetooth Low-Energy Beacon Resistance to Jamming Attack. In *2023 IEEE 13th Int. Conf. on Electronics and Information Technologies (ELIT)* (pp. 270–274).
5. TajDini, M., Sokolov, V., & Buriachok, V. (2019). Men-in-the-Middle Attack Simulation on Low Energy Wireless Devices using Software Define Radio. In *8th Int. Conf. on "Mathematics. Information Technologies. Education:" Modern Machine Learning Technologies and Data Science*, vol. 2386 (pp. 287–296).
6. Pakanon, N., Chamchoy, M., & Supanakoon, P. (2020). Study on Accuracy of Trilateration Method for Indoor Positioning with BLE Beacons. In *2020 6th Int. Conf. on Engineering, Applied Sci. and Technol.* https://doi.org/10.1109/ICEAST50382.2020.9165464
7. Ghamari, M., Villeneuve, E., Soltanpur, C., Khangosstar, J., Janko, B., Sherratt, R., & Harwin, W. (2018). Detailed Examination of a Packet Collision Model for Bluetooth Low Energy Advertising Mode. *IEEE Access, 6*, 46066–46073. https://doi.org/10.1109/access.2018.2866323
8. Kindt, P., Park, S., & Chakraborty, S. (2018). Fast Collision Simulation for Cyclic Wireless Protocols. *ArXiv*. https://doi.org/10.48550/arXiv.1807.01645
9. Raj, M., & Achuthan, K. (2022). Transmission Modeling and Attack Simulation of Bluetooth Low Energy. In *2022 2nd Int. Conf. on Interdisciplinary Cyber Physical Systems (ICPS)* (pp. 93–98). https://doi.org/10.1109/icps55917.2022.00025
10. Shavelis, R., & Ozols, K. (2020). Bluetooth Low Energy Wireless Sensor Network Library in MATLAB Simulink. *J. Sens. Actuator Netw., 9, 38*. https://doi.org/10.3390/jsan9030038
11. Shan, G., Lee, B., Shin, S., & Roh, B. (2017). Design and Implementation of Simulator for Analysis of BLE Broadcast Signal Collision. In *2017 Int. Conf. on Information Networking (ICOIN)* (pp. 448–452). https://doi.org/10.1109/icoin.2017.7899533
12. Nikodem, M., & Bawiec, M. (2019). Experimental Evaluation of Advertisement-based Bluetooth Low Energy Communication. *Sensors, 20*. https://doi.org/10.3390/s20010107
13. Nikodem, M., Słabicki, M., & Bawiec, M. (2020). Efficient Communication Scheme for Bluetooth Low Energy in Large Scale Applications. *Sensors, 20*. https://doi.org/10.3390/s20216371
14. Pang, B., Claeys, T., Vankeirsbilck, J., T'Jonck, K., Hallez, H., & Boydens, J. (2023). A Novel Model to Quantify the Impact of Transmission Parameters on the Coexistence Between Bluetooth Low Energy Pairs. *IEEE Internet Things J., 10*, 1733–1745. https://doi.org/10.1109/jiot.2022.3209874
15. Pang, B., T'Jonck, K., Claeys, T., Pissoort, D., Hallez, H., & Boydens, J. (2021). Bluetooth Low Energy Interference Awareness Scheme and Improved Channel Selection Algorithm for Connection Robustness. *Sensors, 21*. https://doi.org/10.3390/s21072257
16. Kim, M. (2017). An Analysis of Packet Collision Probability due to Inter-Piconet Interference in the Bluetooth Low Energy Networks. *J. Inst, Electr. Eng. Korea, 54*, 3–11. https://doi.org/10.5573/ieie.2017.54.8.3
17. Del-Valle-Soto, C., Valdivia, L., Velázquez, R., Del-Puerto-Flores, J., Varela-Aldás, J., & Visconti, P. (2025). Adaptive Jamming Mitigation for Clustered Energy-Efficient LoRa-BLE Hybrid Wireless Sensor Networks. *Sensors, 25*. https://doi.org/10.3390/s25061931
18. Park, W., Eo, S., Joo, C., & Bahk, S. (2024). B-Hop: Time-Domain Adjustment of BLE Frequency Hopping Against Wi-Fi Beacon Interference. *IEEE Internet Things J., 11*, 12853–12863. https://doi.org/10.1109/jiot.2023.3337252
19. Sokolov, V., Skladannyi, P., & Platonenko, A. (2022). Video Channel Suppression Method of Unmanned Aerial Vehicles. In *2022 IEEE 41st International Conference on Electronics and Nanotechnology (ELNANO)* (pp. 473–477). https://doi.org/10.1109/elnano54667.2022.9927105
20. Sokolov, V., Skladannyi, P., & Korshun, N. (2023). ZigBee Network Resistance to Jamming Attacks. In *2023 IEEE Int. Conf. on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo)* (pp. 161–165). https://doi.org/10.1109/ukrmico61577.2023.10380360
21. Sokolov, V., Skladannyi, P., & Mazur, N. (2023). Wi-Fi Repeater Influence on Wireless Access. In *2023 IEEE 5th Int. Conf. on Advanced Information and Communication Technologies (AICT)* (pp. 33–36). https://doi.org/10.1109/aict61584.2023.10452421
22. Sokolov, V., Skladannyi, P., & Platonenko, A. (2023). Jump-Stay Jamming Attack on Wi-Fi Systems. In *2023 IEEE 18th Int. Conf. on Computer Science and Information Technologies (CSIT)* (pp. 1–5). https://doi.org/10.1109/csit61576.2023.10324031

23. Sokolov, V., Skladannyi, P., & Astapenya, V. (2023). Wi-Fi Interference Resistance to Jamming Attack. In *2023 IEEE 5th Int. Conf. on Advanced Information and Communication Technologies (AICT)* (pp. 1–4). https://doi.org/10.1109/aict61584.2023.10452687
24. Novytskyi, A., Sokolov, V., Kriuchkova, L., & Skladannyi, P. (2025). Determining the Error Distribution of BLE Beacons at Antenna Near and Far Fields. In *4th Int. Conf. on Cyber Hygiene & Conflict Management in Global Information Networks (CH&CMiGIN'2025)*, vol. 4024 (pp. 133–143).