Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри кібербезпеки та захисту інформації
_____ Іван ПАРХОМЕНКО
«____» червня 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)
спеціальність 125 Кібербезпека
(код і назва спеціальності)
освітній ступень бакалавр
освітня програма Кібербезпека
(назва освітньо-професійної програми)
на тему: «Метод захисту CAN-шини»

Виконавець: студент IV курсу, групи КБ-41

_____ **Євгеній РОГОЗА**
(підпис) (ім'я, прізвище)

	Підпис	**Ім'я, прізвище**
Керівник		Олександр ЛАПТЄВ
Нормоконтроль		Сергій ДАКОВ

Київ 2025

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«29» листопада 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньо-професійної програми)

Студенту **КБ-41** **Рогозі Євгенію Олександровичу**
(група) (прізвище ім'я по батькові)

Тема кваліфікаційної роботи Метод захисту CAN-шини

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №6 від 28.11.2024 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Топологія та параметри CAN-шини, характеристики електронних блоків управління (ECU)

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Необхідно ознайомитися з особливостями архітектури CAN-шини, вразливостями її протоколу, дослідити існуючі методи атак і захисту, обрати доцільний метод захисту CAN-шини, провести аналіз його ефективності та розробити рекомендації щодо впровадження у транспортних або промислових системах.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Рекомендації щодо вибору та впровадження ефективного методу захисту CAN-шини, який враховує особливості її архітектури, загрози інформаційній безпеці та можливості інтеграції із сучасними системами моніторингу і виявлення атак.

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 29 листопада 2024 року

Завдання видав _____ (підпис) Олександр ЛАПТЄВ (ім'я, прізвище)

Завдання прийняв до виконання _____ (підпис) Євгеній РОГОЗА (ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	29.11.2024 – 20.01.2025	*виконано*
2	Аналіз літератури	21.01.2020 – 17.02.2025	*виконано*
3	Обґрунтування вибору рішення	18.02.2025 – 21.02.2025	*виконано*
4	Виконати аналітичний огляд атак	23.02.2025 – 14.04.2025	*виконано*
5	Проаналізувати методики виявлення атак.	15.04.2025 – 29.04.2025	*виконано*
6	Розробка рекомендацій щодо підвищення захисту CAN-шини.	30.04.2025 – 28.05.2025	*виконано*
8	Оформлення пояснювальної записки	29.05.2025 – 02.06.2025	*виконано*
9	Підготовка до захисту кваліфікаційної роботи	03.06.2025 – 13.06.2025	*виконано*

Завдання видав _____ (підпис) Олександр ЛАПТЄВ (ім'я, прізвище)

Завдання прийняв до виконання _____ (підпис) Євгеній РОГОЗА (ім'я, прізвище)

Термін подання кваліфікаційної роботи до ЕК 13 червня 2025 року

РЕФЕРАТ

Кваліфікаційна робота складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел, має 64 сторінки основного тексту, 1 таблицю та 35 рисунків. Список використаних джерел містить 33 найменування і займає 4 сторінки.

Метою даної роботи є розробка апаратно-програмних рекомендацій захисту CAN-шини з урахуванням сучасних загроз з метою підвищення рівня кібербезпеки об’єктів. Для досягнення поставленої мети необхідно вирішити такі завдання:

- розглянути архітектуру CAN-шини та її вразливості
- проаналізувати існуючі методи захисту мережі CAN
- вивчити сучасні апаратні та програмні рішення для безпеки CAN
- розробити апаратно-програмні рекомендацій захисту CAN-шини

Методи дослідження кваліфікаційної роботи:

- аналіз літератури
- порівняння існуючих методів захисту
- моделювання та експериментальні дослідження

Об’єктом дослідження є процеси захисту CAN-шини.

Предметом дослідження є методи захисту CAN-шини.

У роботі проведено вивчення сучасних методів захисту CAN-шини, включаючи фізичне сегментування, застосування CAN-фаєрволів, систем виявлення аномалій, резервування каналів, а також логування.

Розроблені апаратно-програмні рекомендації можуть використовуватися розробниками вбудованих систем для підвищення захищеності автомобільних контролерів та інших пристроїв, що працюють із CAN-шиною.

Ключові слова: CAN-шина, захист мережі, безпека автомобільних систем, CAN-фаєрвол, виявлення аномалій, логування.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

CAN	–	Controller Area Network
ECU	–	Electronic Control Unit (Електронний блок управління)
PHY	–	Physical Layer (Фізичний рівень)
DoS	–	Denial of Service (Атака відмови в обслуговуванні)
IDS	–	Intrusion Detection System (Система виявлення вторгнень)
IPS	–	Intrusion Prevention System (Система запобігання вторгненням)
AUTOSAR	–	AUTomotive Open System ARchitecture
CAN-FD	–	CAN with Flexible Data-Rate
J1939	–	Протокол зв’язку у вантажних автомобілях (SAE J1939)
LAN	–	Local Area Network (Локальна мережа)
TLS		Transport Layer Security (Протокол захищеного зв’язку)
PDU		Protocol Data Unit (Протокольна одиниця даних)

ВСТУП

Актуальність даної роботи обумовлена зростаючою роллю автомобільних мереж у сучасних транспортних засобах та збільшенням загроз їх безпеці. Controller Area Network (CAN) є однією з найбільш поширених шин для обміну даними між електронними блоками управління (ECU) в автомобілях, проте її традиційна архітектура не передбачає вбудованих механізмів захисту від кіберзагроз. З огляду на зростання кількості атак на автомобільні мережі, розробка ефективних методів захисту CAN-шини стає надзвичайно важливою для забезпечення безпеки, надійності та цілісності систем управління транспортних засобів.

Метою даної роботи є розробка апаратно-програмних рекомендацій захисту CAN-шини з урахуванням сучасних загроз з метою підвищення рівня кібербезпеки об’єктів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- розглянути архітектуру CAN-шини та її вразливості
- проаналізувати існуючі методи захисту мережі CAN
- вивчити сучасні апаратні та програмні рішення для безпеки CAN
- розробити апаратно-програмні рекомендацій захисту CAN-шини

Об’єктом дослідження є процеси захисту CAN-шини.

Предметом дослідження є методи захисту CAN-шини.

Методи дослідження, використані у роботі:

- аналіз наукової та технічної літератури
- порівняльний аналіз існуючих рішень
- узагальнення практичних аспектів впровадження захисних механізмів у CAN

РОЗДІЛ 1
АНАЛІЗ ТА ОСОБЛИВОСТІ CAN-ШИНИ

1.1. Протокол CAN

1.1.1. Controller Area Network: історія створення та розвитку

Що взагалі таке CAN-шина (CAN-bus)? Це високонадійна послідовна шина зв'язку, яка дозволяє виконавчим елементам та датчикам обмінюватися повідомленнями без центрального контролера. Але як і чому вона була створена?

На початку 1980-х років автомобільна електроніка почала стрімко ускладнюватися. Зросла кількість електронних блоків управління (ECU), а традиційні з'єднання призводили до величезної кількості проводки, що мала високу собівартість та невисоку надійність. І саме в цей час інженери з двох компаній: Robert Bosch GmbH (Рис. 1.1) та Mercedes-Benz (Рис. 1.2) шукали надійний протокол, що підходив би під потреби індустрії, проте такого не знайшлось.

Рисунок 1.1 – Логотип компанії Robert Bosch GmbH

Рисунок 1.2 – Логотип компанії Mercedes-Benz

Тому у 1983 році, під керівництвом Уве Кьонке (Uwe Kiencke), в Bosch офіційно стартували роботи над новим мережним протоколом CAN. Головною мотивацією для його створення було додавання нових функціональних можливостей. Зменшення обсягу проводки було позитивним «побічним ефектом». Почались роботи, коли була закладена архітектура протоколу з використанням NRZ-кодування (Non Return to Zero encoding) та бітового арбітражу. Протягом 1985 року команда Bosch завершила розробку специфікації низькорівневих механізмів обміну даними в мережі та подала патентну заявку на технологію CAN. Вольфґанґ Борст (Wolfgang Borst), Вольфґанґ Ботценгард (Wolfgang Botzenhard), Отто Карл (Otto Karl), Гельмут Шеллінґ (Helmut Schelling) і Ян Унру (Jan Unruh) займались реалізацією механізмів виявлення помилок.

В лютому 1986 року Controller Area Network було представлено на щорічному конгресі Society of Automotive Engineers (SAE) (Рис. 1.3) в Детройті. Ця подія вважається «народженням» протоколу CAN. Його презентували Уве Кьонке, Зігфрід Дайс (Siegfried Dais) і Мартін Літшель (Martin Litschel). Назву ж Controller Area Network мережному протоколу дав Професор Вольфгард Лавренц (Dr. Wolfhard Lawrenz).

Рисунок 1.3 – Логотип Society of Automotive Engineers

В 1987 році, на два місяці раніше запланованого терміну, Intel представив перший контролер CAN 82526 (Рис. 1.5), що стало першим апаратним втіленням протоколу на рівні інтегральних схем (пізніше його замінив 82527). Незабаром

Philips Semiconductors випустила свій контролер 82C200 (Рис. 1.4). Контролер від компанії Intel мав FullCAN імплементацію, Philips же використав імплементацію BasicCAN. FullCAN вимагав меншої потужності від під'єднаних мікроконтролерів, але обмежував кількість одночасно прийнятих фреймів. BasicCAN не мав такої проблеми та потребував менше кристальної площі кремнію для виробництва. Разом ці два чіпи надали можливість проводити практичні тести та почати впровадження CAN у автомобільній промисловості.

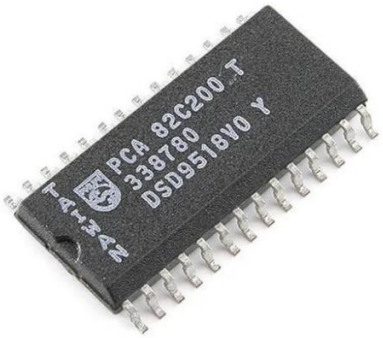

Рисунок 1.4 – Контролер 82C200 створений компанією Philips

Рисунок 1.5 – Контролер 82526 створений компанією Intel

Зараз терміни BasicCAN та FullCAN відійшли в минуле, адже в сучасних версіях протоколу використовується поєднання цих двох підходів.

Вже на початку 1990-х років компанія Bosch розпочала процес стандартизації своєї розробки. Після декількох політичних суперечок, в листопаді 1993 було опубліковано стандарт ISO 11898, а вже через два роки – в

1995, стандарт було доповнено додатком, що описував розширений формат кадру з 29-бітним ідентифікатором CAN.

І все ж, цей стандарт виявився недостовірним, тому заради запобігання несумісним реалізаціям протоколу CAN, компанія Bosh контролювала, щоб усі CAN-чіпи відповідали еталонній моделі CAN від самої Bosch.

Вже в 2003 році Міжнародна організація зі стандартизації (International Organization for Standardization, ISO) публікує переглянуті стандарти Controller Area Network, а саме ISO 11898-1, що визначає «канальний рівень даних CAN» (CAN data link layer), ISO 11898-2, що стандартизує «нестійкий до збоїв, фізичний рівень CAN HS (high-speed)» та ISO 11898-3, що описує «стійкий до збоїв, фізичний рівень CAN LS (low-speed)».

Попри те, що стандарт CAN розроблявся для використання в легкових автомобілях, перші його застосування були іншими. Так фінський виробник ліфтів Kone, використовував вбудовані мережі на основі CAN. А шведське інженерне бюро Kvaser запропонувало протокол CAN для зв'язку в машинах виробникам текстильного обладнання Lindauer Dornier та Sulzer, а також їхнім постачальникам. Тоді під керівництвом Ларса-Берно Фредрікссона (Lars-Berno Fredriksson) з компанії Kvaser ці компанії заснували "CAN Textile User's Group". Вже на початку 1990-х років вони сформували середовище розробки "CAN Kingdom".

В той же час, в Нідерландах, компанія Philips Medical Systems вирішила застосувати його для внутрішньої мережі своїх рентгенівських апаратів та приєдналася до промислових користувачів CAN. Таким чином "Philips Message Specification" (PMS), здебільшого розроблена Томом Сутерсом (Tom Suters), стала першим прикладним шаром для мереж CAN.

В січні 1992 року Хольгер Цельтвандер (Holger Zeltwanger), тодішній редактор журналу VMEbus, зібрав разом користувачів і виробників, щоб створити нейтральну платформу для технічного вдосконалення і маркетингу

CAN. За два місяці була офіційно заснована міжнародна група користувачів і виробників “CAN in Automation” (CiA).

Ще одним академічним підходом займалася Німецька асоціація сільськогосподарських транспортних засобів. Починаючи з кінця 1980-х років вони розробляли шину на основі CAN для сільськогосподарських машин. Але ще до завершення роботи міжнародний комітет вирішив на користь американського рішення J1939 (ISO 11783). Його теж засновано на основі CAN, комітетом SAE Truck and Bus Association.

Стандартизація CAN також відбувалася для вантажівок. Мережеве з’єднання між вантажівкою та причепом стандартизовано як ISO 11992. Цей протокол базується на J1939 та є обов’язковим до використання в Європі з 2006 року.

З 1991 року Mercedes-Benz використовує CAN у своїх пасажирських автомобілях вищого класу. Першим кроком було з’єднання електронних блоків керування (ECU) системою управління двигуном через CAN. У 1995 році BMW впровадила в серії 7 топологію «дерево/зірка» з п’ятьма ECU. Наступним етапом стало підключення блоків для кузовної електроніки. Було реалізовано дві фізично розділені CAN-мережі, часто поєднані через шлюзи. Інші автовиробники наслідували приклад своїх колег зі Штутгарта й зазвичай впроваджували по дві CAN-мережі в своїх легковиках. Нині майже всі виробники обладнують автомобілі кількома незалежними CAN-сегментами.

На початку 2011 року компанії General Motors і Bosch розпочали роботу над покращеннями протоколу CAN з метою збільшення пропускної здатності. Особливо складною була процедура завантаження дедалі більших пакетів програмного забезпечення в електронні блоки керування (ECU). Ця операція займала забагато часу, тож її потрібно було прискоротити за рахунок більш продуктивної системи зв’язку. Ідея збільшити швидкість передачі в CAN шляхом введення другого бітрейту не була новою: ще з початку 2000-х кілька науковців публікували відповідні підходи. Проте жоден із них не був достатньо зрілим, щоб переконати автовиробників. У співпраці з іншими фахівцями Bosch попередньо

розробила специфікацію CAN FD, яка була офіційно представлена в 2012 році на 13-й міжнародній конференції CAN у замку Гамбах (Німеччина).

Доктор Марк Шрайнер (Mark Schreiner) з Daimler надав багато порад і «хитрощів» для проєктування мереж CAN FD.

1.2. Архітектура CAN-шини

1.2.1. Фізичний рівень (PHY)

Фізичний рівень шини CAN реалізовано у вигляді двожильної диференційної лінії (CAN_H і CAN_L), що зв’язує всі вузли у топології шина. Диференційна передача забезпечує стійкість до електромагнітних перешкод: в домінантному стані (логічний 0) напруга в проводі CAN_H піднімається приблизно на +3.5 В відносно маси, в CAN_L опускається до ≈1.5 В, даючи різницю близько +2 В. У рецесивному стані (логічна 1) обидві лінії працюють на проміжній напрузі ≈2.5 В. Для запобігання викривленню сигналів, на кінцях встановлюються термінувальні резистори по 120 Ω, у результаті чого імпеданс шини становить ≈60 Ω загалом.

Трансивери, які підключаються до лінії, виконують кілька ключових функцій: вони перетворюють логічні рівні TTL/CMOS від мікроконтролера в диференційні сигнали шини й навпаки, забезпечують захист від коротких замикань, електростатичних розрядів (ESD) та високочастотних шумів. Сучасні трансивери підтримують інтерфейс standby для зменшення енергоспоживання в стані простою, мають вбудовані схемні фільтри й світлодіодні індикації активності шини (Wake-up).

Максимальна довжина CAN-шини і швидкість передачі пов’язані з абсорбцією й розсіюванням сигналу у кабелі. Стандартний високошвидкісний CAN (HS-CAN) підтримує до 1 Мбіт/с на відстані до 40 м, з пониженням швидкості до 125 кбіт/с довжина шини може досягати 500-1000 м з використанням екранованих кабелів і повторювачів. У CAN FD фізичний рівень

модифіковано для двофазної передачі: у фазі арбітражу (Arbitration Phase) зберігаються класичні параметри, а в фазі даних (Data Phase) швидкість підвищена до 8 Мбіт/с завдяки адаптації порогів прийому й зміни ємності лінії.

Для оптимізації трасування й зменшення електромагнітних випромінювань застосовують кабелі з екраном із заземленням на одному кінці, а в критичних випадках оптичні або гібридні сегментатори із конверторами рівнів. Стандарт ISO 11898-2 також описує рекомендації щодо розставляння шунтів і вибору відповідного перерізу провідників, що критично для довжини понад 100 м.

1.2.2. Канальний рівень (Data Link Layer)

Канальний рівень CAN-шини забезпечує форматування кадрів, арбітраж, виявлення помилок і відновлення передачі без участі вищих рівнів протоколу. Стандарт ISO 11898-1 визначає чотири типи кадрів: Data Frame, Remote Frame, Error Frame та Overload Frame .

Для синхронізації без зовнішньої тактової лінії застосовується механізм bit stuffing: після передачі п’яти послідовних біт однакового рівня передавач вставляє “stuff” біт протилежного рівня, гарантуючи регулярні переходи для ресинхронізації приймачів. Приймач видаляє “stuff” біти перед подальшою обробкою, а поява шести однакових біт у кадрі розглядається як порушення і призводить до генерації Error Frame .

Арбітраж виконується в полі Arbitration за принципом NRZ wired-AND: усі вузли починають одночасно після SOF, але ті, що передають домінантні біти (0), залишаються активними, а вузли, які передавали рецесивні (1), миттєво зупиняються, щойно виявляють 0 на шині. Це забезпечує безвтратний арбітраж та гарантовану доставку повідомлень з вищим пріоритетом, визначеним меншим числовим значенням ID.

Контроль помилок здійснюється через CRC, ACK та Error Counters (Transmit Error Counter і Receive Error Counter). При перевищенні порогів вузол переходить у стани “error passive” або “bus-off” і може відновити роботу лише

після програмного скидання. Механізм автоматичної повторної передачі дозволяє відновити втрачений кадр у наступному доступному інтервалі без участі вищих рівнів, що є критичним для застосувань із жорсткими вимогами до часу.

1.3. Формати кадрів та механізм арбітражу

У мережі CAN всі повідомлення називаються кадрами, і стандарт передбачає чотири їх типи: data frame (дані), remote frame (запит), error frame (помилка) та overload frame (затримка). Найбільше значення має data frame, який використовується для передачі корисних даних між вузлами (Рис. 1.6).

Кожен data frame починається з Start of Frame (SOF) – одного домінантного біту (логічний 0), що позначає початок передачі. Після SOF іде арбітражне поле (Arbitration Field): тут передаються ідентифікатор повідомлення та біт RTR (Remote Transmission Request). Ідентифікатор у стандартному форматі має 11 біт (CAN 2.0A), а в розширеному – 29 біт (CAN 2.0B) завдяки додатковому полю Extension ID. Біт IDE (Identifier Extension) визначає формат кадру: домінантний для стандартного, рецесивний для розширеного.

Далі йде Control Field, яке містить біт r0 (зарезервований) та чотирьохбітове поле DLC (Data Length Code), що визначає довжину поля даних від 0 до 8 байт у Classic CAN і до 64 байт у CAN FD. За ним слідує безпосередньо Data Field із самими байтами даних.

Після даних йде Cyclic Redundancy Check (CRC) Field: у Classic CAN це 15-бітний код плюс 1-бітний CRC Delimiter, який рецесивний для відокремлення поля CRC від ACK Field. Наступне – ACK Field: передається двома бітами, де перший (ACK Slot) відправник резервує як рецесивний (1), а будь-який приймач, що прийняв кадр без помилок, замінює його на домінантний (0) для підтвердження. Закінчує кадр End of Frame (EOF) сім рецесивних біт, після чого йде мінімум три рецесивні біти Інтерфреймового Простору (Interframe Space), що розділяють кадри.

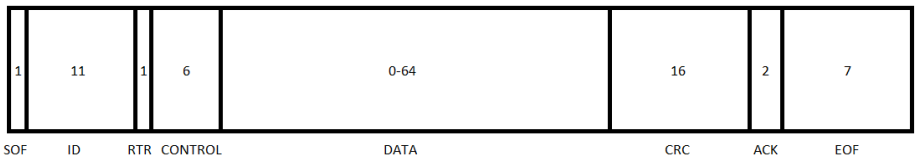

Рисунок 1.6 – Формат кадру CAN

Щоб забезпечити синхронізацію без окремого тактового сигналу, усі біти, крім CRC Delimiter, ACK Field та EOF, піддаються bit stuffing: після п’яти підряд біт однакового рівня вставляється біт протилежного рівня. Це гарантує достатню кількість переходів для коректної синхронізації приймача й передавача. При відновленні даних отримувач видаляє вставлені “stuff” біти.

Арбітраж в CAN є безвтратним: всі вузли починають передавати одночасно після SOF. Перший біт – SOF, що домінантний, у наступні біти ідентифікатора вузли з вищим пріоритетом (нижчим числовим значенням ID) передають більше домінантних (0) бітів. Якщо вузол передає рецесивний (1), але бачить на шині домінантний (0), він одразу припиняє передачу, але без генерування помилки, й очікує наступного вільного вікна для повтору.

Наприклад, два вузли з ID 15 (00000001111) та 16 (00000010000) починають одночасну передачу. Перші вісім біт ідентифікатора ідентичні (00000001), арбітраж не відбувається. На дев’ятому біті вузол 16 передає "1”, бачить “0” та зупиняється, а вузол 15 продовжує передачу без втрат.

Такий механізм забезпечує гарантію передачі та пріоритетність повідомлень без конфліктів і втрат даних.

1.4. Вразливості та загрози для CAN

1.4.1. Перехоплення та модифікація повідомлень

В традиційній архітектурі CAN-шини повністю відсутні механізми аутентифікації чи шифрування повідомлень, тому всі кадри передаються у відкритому вигляді та можуть бути перехоплені та змінені без тривалих зусиль.

Пасивне перехоплення (sniffing). Будь-який пристрій із фізичним доступом до CAN-інтерфейсу, наприклад, через діагностичний OBD-II порт, може без перешкод зчитувати ідентифікатори та вміст кадрів, не впливаючи на роботу мережі. Такі dongle-пристрої (Рис. 1.7) (наприклад, дешеві Bluetooth-адаптери) під'єднуються до OBD-II і забезпечують повний "sniffing" внутрішніх сигналів шини (драйвтрайн, кузовна електроніка тощо) без додаткових привілеїв. Вивчивши перехоплені повідомлення, зловмисник може побудувати детальну карту взаємодії ECU, визначити періоди опитування та пріоритетність вузлів, що суттєво спрощує підготовку до цілеспрямованих атак.

Рисунок 1.7 – OBD-II донгл

Активна модифікація та ін'єкція кадрів. Після перехоплення даних зловмисник може відправляти у шину власні повідомлення (спуфінг) або прямо модифікувати поле даних. У дослідженні Purdue University показано, що ін'єктовані повідомлення з низьким ID блокують передачу кадрів з вищим пріоритетом, викликаючи Denial-of-Service на рівні окремих ECU. Спуфінг дозволяє змінювати керуючі команди (наприклад, кермо, гальма, швидкість) без жодного сповіщення про помилку на блоці управління.

Bit-level атаки з FPGA-маніпуляторами. Сучасні FPGA-рішення (FPGA-Based Low-level CAN Protocol Testing) реалізовують “man-in-the-middle” атаки на фізичному рівні: вони перехоплюють диференційні сигнали CAN_H/CAN_L, змінюють потрібні біти між фазою арбітражу й даних та одночасно перераховують CRC, щоб вузли-отримувачі не помітили порушення протоколу. Затримка між прийомом і повторною передачею кадру може бути менше 100 нс, що гарантує успішне проходження арбітражу та приховує втручання.

Такі FPGA-маніпулятори можуть не лише ін’єктувати спуфінгові повідомлення, а й на льоту підміняти окремі біти, наприклад змінюючи значення сенсорів чи команд керування.

Традиційні IDS/IPS для CAN-шини, які аналізують лише інтервали та шаблони трафіку, часто не реагують на тонкі модифікації вмісту кадрів, що не змінюють їхню довжину чи періодичність. Реальні кейси показують, що атаки на рівні ID та даних можуть тривати непоміченими до моменту виникнення критичної ситуації.

1.4.2. Атаки типу Denial-of-Service

Атаки типу Denial-of-Service (DoS) на шину CAN є одними з найпростіших для реалізації й водночас найнебезпечніших, оскільки вони призводять до втрати зв’язку між електронними блоками керування (ECU) та блокують нормальний обмін даними. У основі DoS-атаки лежить експлуатація механізму безвтратного арбітражу CAN, де повідомлення з найнижчим числовим ідентифікатором (ID) мають вищий пріоритет.

Найпоширенішою формою DoS-атаки є bus-flooding: зловмисник підключає недорогу мікроконтролерну плату до шини через OBD-II, Infotainment або будь-який інший CAN-інтерфейс і безперервно передає кадри з ID=0x00 (або іншим мінімальним ID). Оскільки такі кадри завжди «виграють» арбітраж, легітимні вузли не можуть передати жодного повідомлення. Результатом є повний «завис» шини – нульова пропускна здатність для всіх інших ECU.

Відновити передачу даних можна тільки після припинення flood-атаки або перезавантаження всіх пристроїв, що на ходу практично неможливо.

Другий вид DoS-атак використовує fault-confinement механізм CAN: після накопичення певної кількості помилок вузол переходить у стан bus-off і перестає передавати кадри до програмного скидання. Атакуючий вводить помилки bit-stuffing (розміщує домінантні біти там, де мають бути рецесивні) або навмисно передає некоректний CRC в кадрах. Після кількох таких помилок конкретний ECU відключається від шини й перестає працювати, хоча решта мережі продовжує обмінюватися даними. Це дає можливість вивести з ладу окремі підсистеми – наприклад, антиблокувальну систему гальм (ABS) або систему керування двигуном, не впливаючи на інші ECU.

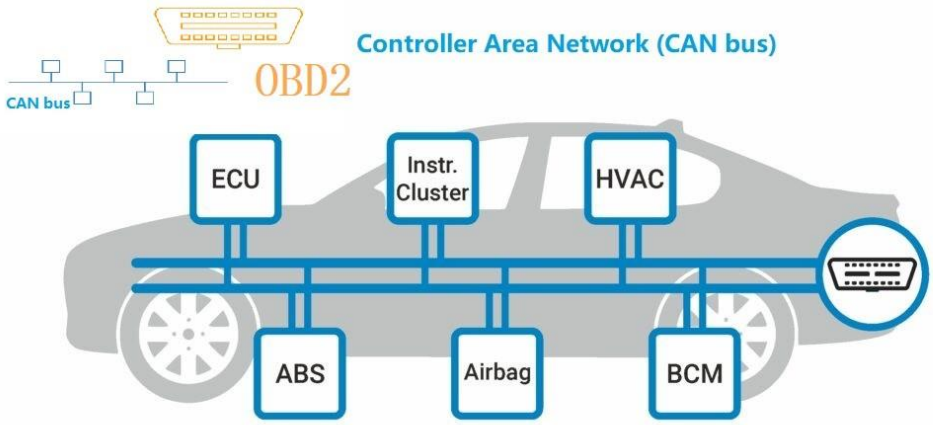

Рисунок 1.8 – Блоки підключені до CAN-шини

Третій варіант – цілеспрямований ECU-Dos. Замість глобального flood-атаки зловмисник спочатку проводить пасивне перехоплення (sniffing), щоб ідентифікувати частоту та час передачі кадрів конкретного вузла. Потім у критичні моменти він ін’єктує Error Frame або кадри з високим пріоритетом, викликаючи у цільовому ECU багаторівневі помилки та перехід у bus-off. Така атака виявляється лише після втрати роботи підсистеми, а сама шина може продовжувати функціонувати в обмеженому режимі.

Найвитонченішим підходом є стелс-DoS, зокрема атаки типу WeepingCAN. Зловмисник розподіляє flood-трафік на триваліший час із низькою частотою помилкових кадрів, щоб ускладнити виявлення за допомогою традиційних IDS, які аналізують різкі сплески трафіку. Поступово цільовий ECU накопичує помилки й також переходить у bus-off без явних ознак атаки.

Для реалізації DoS-атак на фізичному рівні використовують FPGA-базовані маніпулятори сигналів, які перехоплюють диференційний сигнал CAN_H/CAN_L, аналізують фазу арбітражу та передачі даних, а потім на льоту ін’єктують кадри високого пріоритету або некоректні біти з перерахунком CRC. Затримка між прийомом і повторною передачею кадру може становити менше ніж 100 нс, тож інші вузли просто не помічають зміни й обробляють кадр як легітимний.

Небезпека DoS-атак на CAN-шину полягає в тому, що їх можна здійснити без доступу до прошивок ECU, використовуючи лише фізичний або безпровідний доступ через OBD-II чи Bluetooth. Вони створюють прямий ризик для безпеки: від раптової зупинки двигуна та втрати гальм до неможливості управління кермом або відключення критичних систем, таких як подушки безпеки.

1.4.3. Бездротові інтерфейси та розширення шин

У сучасних автомобілях функціональні можливості шини CAN значно розширюються за рахунок підключення до неї бездротових інтерфейсів і зовнішніх мереж. До таких інтерфейсів належать діагностичні OBD-II dongle (Bluetooth, Wi-Fi, LTE), модулі телематики (Telematics Control Unit, TCU), адаптери TPMS (ZigBee) та навіть вбудовані в мультимедійні системи Wi-Fi й Bluetooth. Кожен з них створює додаткову точку входу в мережу, що робить можливими віддалені атаки на CAN-шину без фізичного підключення до OBD-II порту (Рис. 1.9).

Рисунок 1.9 – Роз’єм OBD-II

Здебільшого бездротові OBD-II адаптери служать для зчитування даних про діагностику двигуна та параметри руху, проте відомо про численні вразливості їх програмного забезпечення. Наприклад, дослідні Bluetooth-dongle відкривають незашифровані порти, через які зловмисник може під’єднатися й перехопити трафік CAN у режимі реального часу. У різних тестах показано, що зчитування параметрів і навіть модифікація кадрів через такі dongle можлива без обмежень протягом перших секунд підключення.

Ключовим елементом будь-якої сучасної інфраструктури “connected car” є модуль телематики (TCU) (Рис. 1.10), що забезпечує зв’язок із хмарою та мобільними мережами. Через вразливості в прошивці TCU атакуючі можуть отримати доступ до внутрішньої мережі транспортного засобу, а відтак до CAN-шини. Протоколи, як-от MQTT або HTTPS, які використовуються для обміну даними з хмарою, не гарантують сегрегації CAN-мережі, що й створює можливість ін’єкцій кадрів із зовнішньої мережі без персональної присутності в салоні.

Рисунок 1.10 – TCU для Ford Fiesta

Окрім Bluetooth і LTE, в деяких транспортних засобах застосовуються бездротові датчики тиску в шинах (TPMS), що часто працюють на протоколі ZigBee. У гібридних системах, де CAN-шина розмикається та доповнюється бездротовою лінкою, демонструвалися DoS-атаки шляхом джемінгу ZigBee-каналу. Такі атаки призводять до втрати зв’язку між сенсорами TPMS і центральним блоком, але можуть бути масштабовані на інші бездротові сегменти шини, що використовують ті самі частоти й топології.

Наукові дослідження також звернули увагу на Wi-Fi-мережі, вбудовані в інфотейнмент-системи. Уразливості в цих мережах дозволяють зловмисникам під’єднатися до медіаблоку й далі отримати доступ до CAN-шини шляхом «містка» між Wi-Fi та Ethernet або USB. Прикладом є атаки, які починаються з експлуатації вразливостей HTML5-інтерфейсу мультимедіа, потім – встановлення шкідливої прошивки в TCU й, нарешті, отримання привілеїв для ін’єкцій кадрів в шину.

Таким чином, розширення CAN-шини бездротовими інтерфейсами суттєво збільшує поверхню атаки, відкриваючи вектори для віддалених вторгнень, DoS-атаки, спуфінгу та ін’єкції кадрів у мережу. Ефективний захист потребує комплексного підходу, що поєднує криптографічні рішення, сегментацію мережі та фізичний контроль доступу.

1.5. Replay attack

В цьому розділі демонструється реалізація replay-атаки на CAN-шину із спуфінгом кадру, що відповідає за сигнал натискання педалі газу в автомобілі, з використанням opensource-інструментів для Linux. Для експерименту застосовувалася програма ICSim у поєднанні з пакетом утиліт can-utils, що дозволяють прослуховувати та ін’єктувати CAN-фрейми. Початковим етапом було встановлення необхідного програмного забезпечення.

Після завантаження та встановлення необхідних пакетів, я запустив симуляцію (Рис. 1.11).

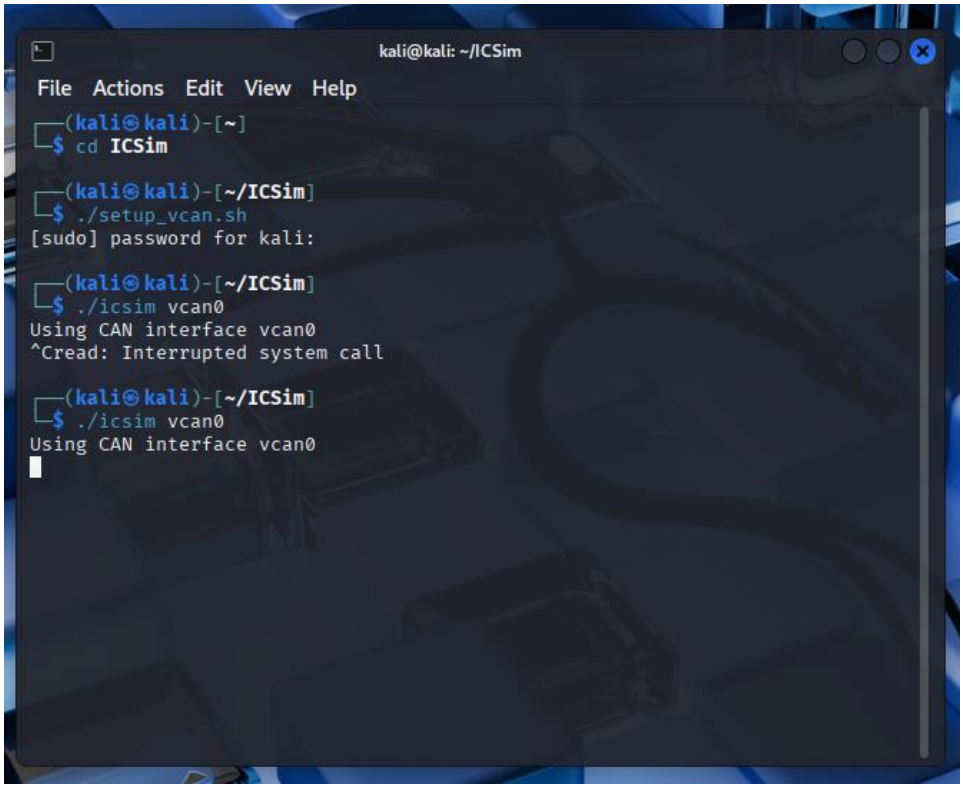

Рисунок 1.11 – Запуск симулятора

Рисунок 1.12 – Відображення роботи симулятора

Після цього запускаємо роботу симулятора (Рис. 1.13).

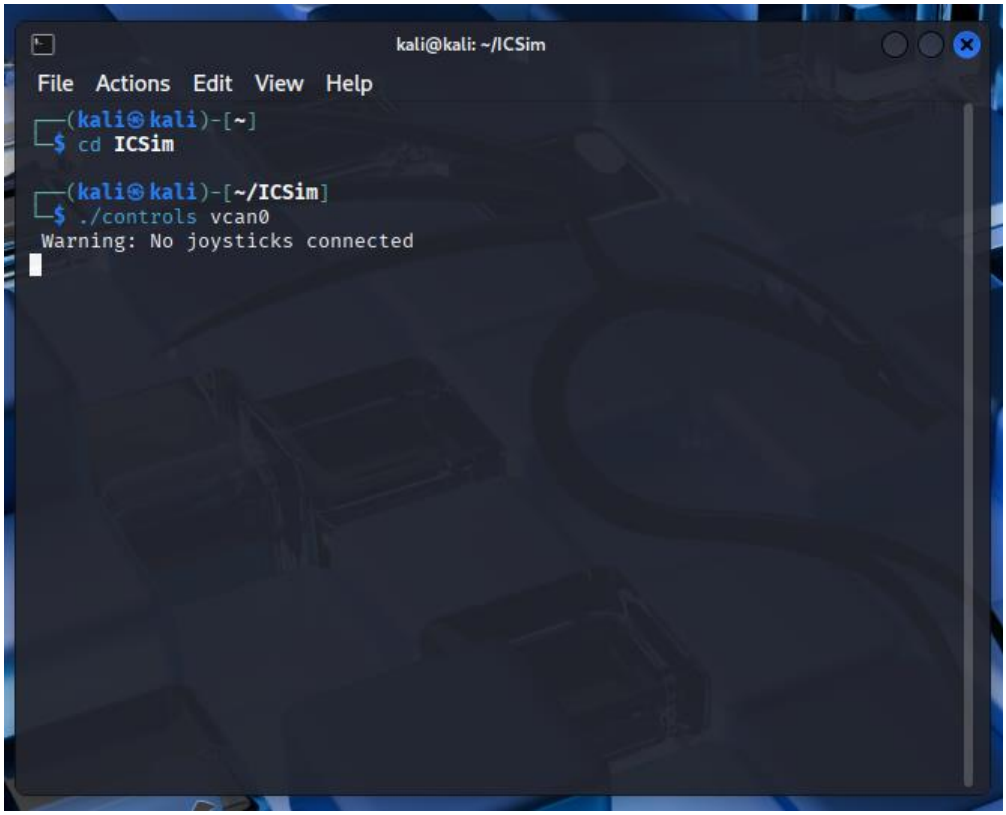

Рисунок 1.13 – Запуск роботи симулятора

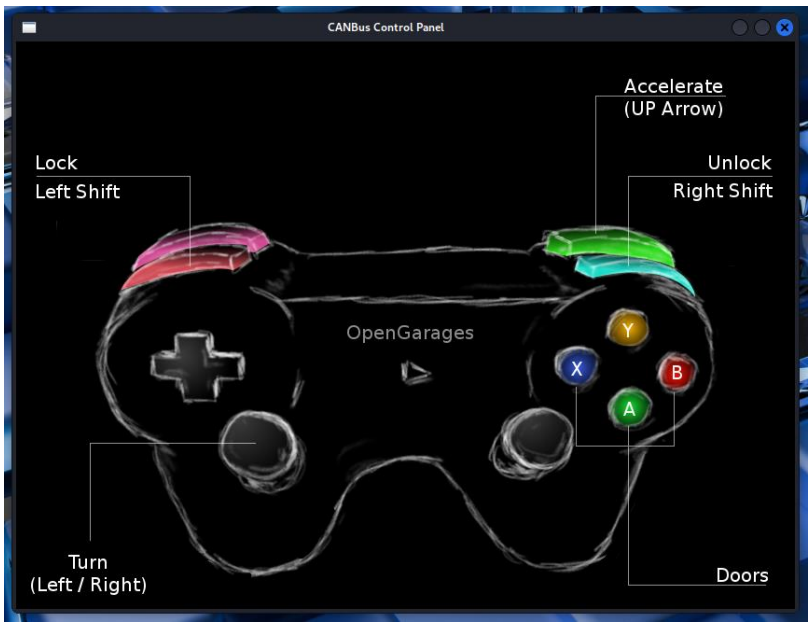

Рисунок 1.14 – Кнопки керування симулятором

Симулятор починає працювати. Ми можемо бачити кадри, що передаються CAN-шиною (Рис. 1.15).

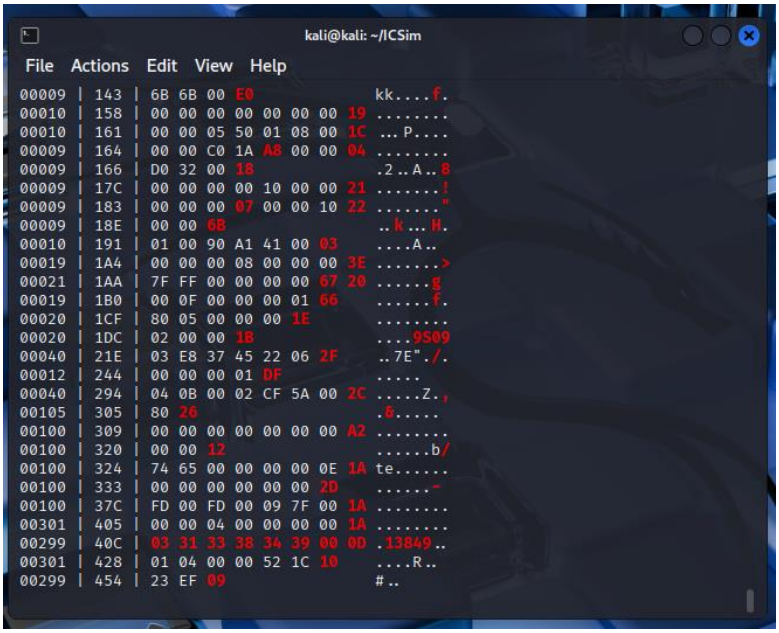

Рисунок 1.15 – Кадри, що передаються CAN-шиною

Після цього, визначивши потрібний кадр (натискання педалі газу), ін’єктуємо кадр до шини (replay attack) (Рис. 1.16).

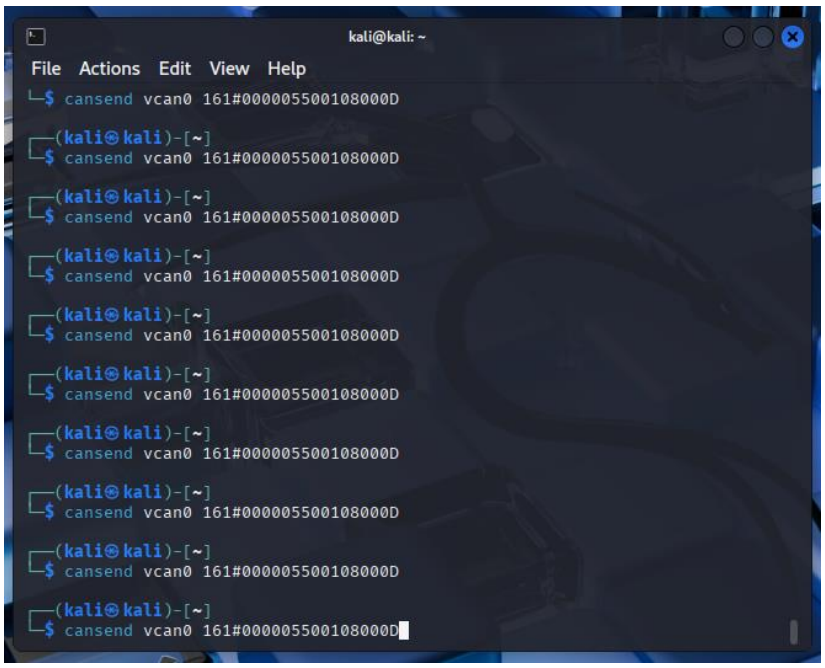

Рисунок 1.16 – Ін’єктуємо кадр

Рисунок 1.17 – Двигун набирає оберти

Двигун набрав оберти – ми успішно провели спуфінг (replay attack) кадру до CAN-шини (Рис. 1.17).

Висновки за розділом 1

У результаті огляду протоколу CAN, його архітектури та форматів кадрів, а також аналізу вразливостей і загроз стає очевидним, що традиційна CAN-шина, попри високу надійність і простоту реалізації, потребує додаткового захисту.

Відсутність аутентифікації та шифрування робить можливим пасивне перехоплення й активну модифікацію повідомлень, що відкриває шлях для спуфінгу та DoS-атак. Зловмисники можуть використовувати недорогі dongle-адаптери або апаратні FPGA-маніпулятори для ін’єкції кадрів і штучного виключення ECU через механізми bus-off.

Поява бездротових інтерфейсів (Bluetooth, Wi-Fi, LTE, ZigBee) і телематичних модулів значно розширила поверхню атаки: порушення у прошивках TCU, вразливості OBD-II адаптерів чи DNS-містки мультимедіа-систем дають віддалений доступ до шини без фізичної присутності. Усе це ставить під загрозу критичні підсистеми авто – від системи кермування до гальмівних комплексів і подушок безпеки.

РОЗДІЛ 2
МЕТОДИ ЗАХИСТУ CAN-ШИНИ

2.1. Фізичне сегментування та ізоляція шини

У класичній схемі всі електронні блоки керування (ECU) підключені до єдиної лінії CAN_H/CAN_L, що спрощує інсталяцію, але водночас створює значний ризик: будь-яка загроза чи збій на одному блоці шини миттєво поширюються на всі вузли і можуть призвести до відмови критичних підсистем. Фізичне сегментування полягає в розподілі загальної CAN-мережі на кілька ізольованих сегментів, які взаємодіють між собою лише через спеціальні шлюзи чи мультиплексори.

Розбиття шини на сегменти дозволяє локалізувати помилки та активні атаки. Наприклад, якщо на діагностичному сегменті, до якого під’єднано OBD-II порт, виникне DoS-атака чи програмований спуфінг-пристрій спробує ін’єктувати шкідливі кадри, такі повідомлення не проникнуть у сегмент, де зосереджені блоки керування гальмами чи подушками безпеки. Апаратура шлюзу переглядає і фільтрує потік, відкидаючи небажані ID або кадри, що не відповідають налаштованим політикам. Одним із прикладів такого рішення є Ixxat CAN@net NT 420 від HMS Networks (Рис. 2.1), який забезпечує двосторонню маршрутизацію між сегментами, підтримує конвертацію швидкостей (CAN 2.0A/B і CAN FD) та виконує гальванічну розв’язку (між двома CAN-сегментами немає прямого електричного зв’язку – сигнал передається через оптичні або індуктивні перетворювачі), запобігаючи поширенню збоїв з одного сегмента на інші.

Рисунок 2.1 – Логотип HMS Networks

Шлюзи типу CAN gateways можуть бути як “hard-wired”, так і програмовані. У першому випадку апаратний пристрій має жорстко закладені правила маршрутизації та фільтрації, у другому – сервісне ПЗ дозволяє змінювати політики на льоту, не відключаючи мережу. Останній підхід особливо корисний під час експлуатації транспортного засобу або виробничої лінії, коли необхідно оперативно розширити чи звузити межі доступу між сегментами.

Крім розділення по лінії CAN, важливо контролювати фізичний доступ до зон, які не повинні бути відкриті для сторонніх пристроїв. Стандартний роз’єм OBD-II легко використати в якості точки атаки: через нього можна підключитися до шини, отримати доступ до діагностичних і сервісних кадрів, а потім ін’єктувати шкідливий трафік. Одним із ефективних методів захисту є встановлення механічних кришок або заслінок на роз’ємі (Рис. 2.2), які відкриваються лише після автентифікації ключем або електричним сигналом. Альтернативою є вбудовування реле, яке живить OBD-II порт лише при виконанні умов безпеки (наприклад, при наявності пароля) — без напруги на CAN_H/CAN_L зовнішній адаптер фізично не під’єднається до шини.

Рисунок 2.2 – Замок на OBD-II порт

Інший варіант – заміна стандартного 16-контактного роз’єму на захищений конектор: частина контактів блокується до моменту процедури автентифікації, а після перевірки легітимності підключення відкривається фізичний доступ до повного набору сигналів. Такі рішення вже пропонують деякі OEM-виробники в рамках проєктів зі стандартом ISO 15118 “Vehicle-to-Grid”, де висока безпека є обов’язковою умовою для зарядки електромобілів.

Після впровадження сегментації мережі та захисту до OBD-II порту важливо ретельно налаштувати правила маршрутизації та фільтрації в шлюзах. Наприклад, кадри з певними ID, що використовуються виключно для внутрішнього обміну між двигуном і трансмісією, слід категорично не пропускати в сегмент діагностики. Аналогічно, повідомлення від телематичного модуля (Telematics Control Unit (TCU)) віддалено підключеного фізично через LTE – мають виконувати лише службові функції (OTA-оновлення, передачу діагностики) і не мати доступу до критичних команд керування.

Важливо зазначити, що фізичне сегментування не впливає на внутрішній протокол CAN: ми не змінюємо структуру кадрів чи механізми арбітражу.

Натомість його завдання забезпечити, щоб повідомлення, які не відповідають політикам безпеки, навіть не потрапляли на сегменти, де ці кадри можуть мати критичні наслідки. Такий підхід також підвищує стійкість системи до електромагнітних перешкод і коротких замикань – гальванічна розв'язка (Рис. 2.3) між сегментами знижує ризик поширення шуму.

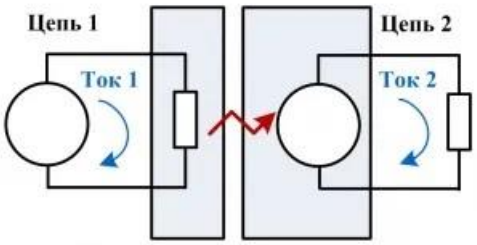

Рисунок 2.3 – Гальванічна розв'язка

Практична імплементація фізичного сегментування потребує ретельного проектування топології мережі. Необхідно аналізувати всі потоки даних між ECUs і визначати, які з них дійсно потребують взаємодії, а які можна відокремити в ізольований сегмент. Після цього вказують у конфігурації шлюзів, які ID та частоти кадрів передавати між сегментами, а які блокувати. Сучасні шлюзи підтримують як статичні правила, так і динамічні фільтри на основі часу та об'єму трафіку, що дозволяє реалізовувати складні механізми rate-limiting без додаткових мікроконтролерів на стороні ECU.

На рівні промислових мереж аналогічні рішення існують у вигляді CAN-to-Ethernet або CAN-to-TSN шлюзів, де сегмент CAN ізолюється від Ethernet-сегменту, з'єднуючись через захищені канали з TLS-шифруванням. У таких сценаріях фізичне сегментування поєднується з криптографічними методами на верхніх рівнях OSI, але сам CAN-шар залишається незмінним.

Загалом, фізичне сегментування та ізоляція CAN-шини – це основа будь-якої стратегії безпеки, спрямованої на захист від атак «з коліс», на відмовостійкість у разі апаратних збоїв та на локалізацію інцидентів. Навіть без глибоких змін у протоколі CAN, цей метод дозволяє забезпечити сегментацію

доступу, усунути єдину точку відмови та значно зменшити поверхню атаки на критичні підсистеми.

2.2. CAN-фаєрволи та шлюзи протоколів

Всі ECU (Electronic Control Units), підключені до однієї лінії CAN_H/CAN_L, сприймають будь-які кадри з правильним CRC та коректною структурою, що дозволяє зловмиснику легко прослуховувати мережу та ін’єктувати шкідливі повідомлення. CAN-фаєрволи та шлюзи протоколів розташовуються між різними функціональними сегментами – наприклад, між діагностичним портом OBD-II та критичними блоками ABS або подушок безпеки і здійснюють багаторівневий аналіз та фільтрацію трафіку на апаратному та програмному рівнях, запобігаючи поширенню небажаних чи шкідливих повідомлень у критичні підсистеми.

Перший рівень захисту забезпечує статична фільтрація, де правила «білого» та «чорного» списків за ID визначають, які кадри обов’язково пропускаються, а які блокуються одразу на вході або виході сегмента. Це найпростіший, але дуже ефективний спосіб уникнути спуфінгу, коли зловмисник підміняє ID легітимного ECU. Динамічна фільтрація (stateful packet inspection) підвищує рівень безпеки, оскільки фаєрвол відстежує послідовність обміну повідомленнями, контролює контекст сеансу та блокує нетипові переходи або неочікувані зміни.

Щоб протидіяти DoS-атакам, в яких мережа «затоплюється» надмірним потоком кадрів з пріоритетними ID, CAN-фаєрволи реалізують rate-limiting — обмеження числа повідомлень за одиницю часу за конкретними ID або від конкретних портів. Якщо ліміт перевищено, фаєрвол починає відмовляти у надлишкових запитах, зберігаючи арбітражну здатність мережі для легітимного трафіку .

Другий рівень захисту впроваджує Deep Packet Inspection (DPI), що аналізує не лише заголовки кадрів, але й вміст полів даних. Це необхідно для

виявлення складних атак, коли змінюють внутрішні байти корисного навантаження, але залишають ID та DLC незмінними, щоб обійти базову фільтрацію. DPI дозволяє визначити невідповідність очікуваному форматному профілю повідомлень – наприклад, коли параметри сенсорів виходять за межі нормальних діапазонів, і заблокувати такі кадри до того, як вони потраплять до ECU.

Третій ключовий компонент – маршрутизація та конвертація швидкостей, яку забезпечують протокольні шлюзи. У багатьох сучасних автомобілях співіснують Classic CAN (до 1 Мбіт/с) та CAN FD (до 8 Мбіт/с), а також сегменти Automotive Ethernet для високошвидкісних функцій (OTA-оновлення, мультимедіа). Протокольний шлюз приймає кадри з одного інтерфейсу, застосовує політики маршрутизації, буферизує їх і передає в інший сегмент з відповідною адаптацією бітрейту та форматів, зберігаючи сумісність з наявними ECU та мінімізуючи необхідність заміни обладнання.

Також між сегментами застосовується гальванічна розв’язка. Вона фізично роз’єднує електричні ланцюги передавального та приймального інтерфейсів, передаючи інформацію через індуктивні, оптичні або конденсаторні перетворювачі. Це запобігає перенесенню шуму і різниць потенціалів між вузлами, що значно підвищує стійкість мережі.

2.3. Системи виявлення аномалій (Physical & Behavioral IDS)

Системи виявлення аномалій (Intrusion Detection Systems, IDS) у мережах Controller Area Network (CAN) поєднують два взаємодоповнювальні підходи – фізичний аналіз сигналів (Physical IDS) та поведінковий моніторинг (Behavioral IDS), щоб забезпечити багаторівневий захист від кібератак і несподіваних збоїв обладнання.

Перший рівень, Physical IDS, ґрунтується на тому, що кожен електронний блок керування (ECU) генерує унікальні електричні характеристики сигналів при передачі байтів по диференційних лініях CAN_H/CAN_L. Fitbit ECU має власний

«відбиток» часових затримок сигналу та амплітудних рівнів, що залежать від характеристики трансивера, довжини проводки та компонентів. Вимірювання годинникових зсувів (clock skew) (Рис. 2.4) дає змогу розпізнати кожен вузол за патерном затримки між режимом очікування шини і реальним стартом передачі кадру . Цей підхід дозволяє навіть при ідентичних ID виявити шкідливий пристрій, який намагається видати себе як легітимний ECU. Аналогічно, VoltageIDS аналізує напругу сигналу, вимірюючи форму верхні та нижні краї диференційного сигналу. Різні моделі трансиверів мають характерні хвилі та невеликі відмінності амплітуди, що використовуються як фізичний «відбиток» (fingerprint).

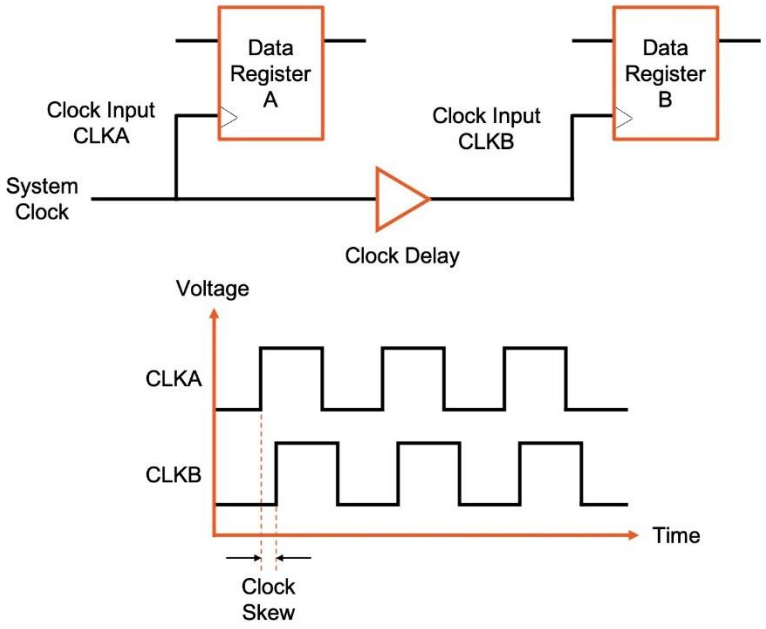

Рисунок 2.4 – Годинниковий зсув (clock skew)

Physical IDS зазвичай складається з апаратного модулю, встановленого між сегментами шини або прямо в Gateway-ECU, що виконує векторний аналіз вхідних сигналів у реальному часі з високою точністю. FPGA-реалізації, такі як FP-IDPS, можуть обробляти тисячі кадрів на секунду з мінімальною затримкою (<100 нс), що необхідно для мереж CAN FD з високою швидкістю даних . Однак метою є не тільки висока продуктивність, а й мінімізація хибних спрацьовувань:

помилкові позитивні спрацьовування можуть призвести до відключення легітимних ECU і непередбачуваних станів автомобіля.

Другий рівень, Behavioral IDS, звертає увагу на логіку та патерни обміну повідомленнями: інтервали між кадрами, статистику частоти, залежності між різними ідентифікаторами (ID). Класичні статистичні підходи базуються на нормальному розподілі інтервалів обміну для кожного ID – якщо повідомлення приходить занадто часто або занадто рідко порівняно з навченим профілем, система фіксує аномалію. Такі методи добре виявляють DoS-флуд, коли мережа заповнюється кадрами з привілейованими ID, а також повторювані атаки (replay).

Сучасні Behavioral IDS активно використовують методи машинного навчання. Unsupervised Autoencoder-моделі навчаються стискати вхідні послідовності кадрів у латентний простір, а потім оцінюють похибку рекомпресії нових даних: великі відхилення сигналізують про невідомі або спотворені послідовності. Архітектура CAN-BERT адаптує трансформери з NLP-сфери: підготовка проводиться на послідовностях арбітражних полів (IDs) із застосуванням маскованого передбачення (masked language modeling), після чого модель у реальному часі прогнозує наступний ID, визначаючи невідповідність як ознаку спроби спуфінгу чи модифікації.

Комбінація Physical та Behavioral IDS створює синергію – фізичний модуль миттєво виявляє нелегальні пристрої навіть за збереженого нормального трафіку, тоді як поведінковий аналіз розпізнає нові та складні сценарії атак, що не мають фізичних відмінностей. Так, поєднуються вимірювання амплітудно-часових характеристик та аналіз DLC й інтервалів, гарантувавши виявлення атак із мінімальною затримкою.

Попри значні переваги, IDS стикаються з викликами – баланс між чутливістю та хибними спрацьовуваннями, адаптація до варіацій в стандартному трафіку (наприклад, зміни через оновлення ПО або відмінності між моделями ECU), необхідність підтримки високих швидкостей CAN FD і великих обсягів даних у DLC до 64 байт. Водночас розвиток апаратних платформ, оптимізованих для ML (Edge TPU, FPGA) і вбудованих HSM, забезпечує можливість реалізації

складних виявляючих алгоритмів без порушення жорстких часових вимог сучасних автомобільних систем.

Узагальнюючи, поєднання Physical та Behavioral IDS у мережі CAN дозволяє створити надійну систему виявлення аномалій, здатну протистояти широкому спектру загроз – від фізичного спуфінгу до складних атак на рівні логіки обміну. Ретельна інтеграція, адаптація порогів, комбінування алгоритмів та централізоване реагування роблять такий підхід одним із найбільш перспективних у галузі автомобільної кібербезпеки.

2.4. Резервування каналів і безпечні стани

Резервування каналів та механізми переходу в безпечні стани відіграють ключову роль у забезпеченні відмовостійкості та функціональної безпеки автомобільних мереж на базі Controller Area Network (CAN). Стандартний протокол CAN не передбачає вбудованих засобів для автоматичного відновлення зв’язку у разі фізичної несправності лінії або апаратного збою, отже необхідне впровадження зовнішніх архітектурних рішень, які гарантують, що жоден окремий елемент не стане єдиною точкою відмови в мережі. Першим і найчастішим методом є прокладання двох паралельних фізичних шин CAN_H/CAN_L до ключових ECU – обидві лінії передають однаковий трафік у режимі гарячого дублювання (hot standby), причому тільки одна з них, основна, обробляється в кожен момент часу. Вхідні трансивери постійно спостерігають обидва канали: якщо відбудеться обрив проводки, коротке замикання або рівень помилок CRC у основній шині перевищує встановлений поріг (наприклад, Transmit/Receive Error Counter > 127), апаратна логіка переключає всі передачі на резервний канал із затримкою, яка за сучасними реалізаціями на FPGA або спеціалізованих ASIC не перевищує 1 мс. У цей же час FIFO-буферизатори накопичують кадри, що надходили в момент перемикання, та потім відтворюють їх у тому самому порядку, забезпечуючи збереження пріоритетності повідомлень,

безперебійності передачі критичних даних для ABS, ESC, EPS та інших підсистем.

Як економічний альтернативний підхід до прокладання другої фізичної шини, у низці проектів використовують існуючу силову лінію живлення автомобіля (12 В або 24 В) як резервний канал. Схеми з LC-фільтрами та coupling-inductors відокремлюють високочастотні імпульси від двигуна та інших потужних навантажень від цифрових сигналів CAN. Коли основний канал виходить із ладу, спеціальний трансивер перемикається на лінію живлення й починає передавати та приймати кадри зі швидкістю до кількасот кбіт/с. Така реалізація забезпечує стійкість до фізичних пошкоджень основної шини без значного підвищення вартості кабельної розводки, хоча й потребує суворого екранування та контролю EMI. Цей метод дозволяє підтримувати зв’язок для важливих діагностичних і телематичних даних, навіть якщо основна шина перебуває у несправному стані.

У межах одного ECU із підвищеними вимогами до безпеки часто застосовують внутрішнє дублювання апаратної платформи. Два окремі мікроконтролери або процесорні ядра виконують роль основного та резервного контролерів – вони синхронізуються за допомогою протоколів TTCAN або FlexRay, при цьому основний керує потоком кадрів, а резервний паралельно приймає їх копії. Якщо первинний контролер переходить у стан bus-off через накопичення помилок або зупиняється через внутрішню помилку програмного забезпечення, резервний безшовно продовжує передавання та обробку кадрів зі збереженням часу передачі та пріоритетності, що забезпечує безперервну роботу двигуна чи гальмівної системи навіть за ураження одного з контролерів .

Кожен із цих підходів доповнюється чітко прописаними алгоритмами перемикання. Після перевищення порога помилок Error Counters трансивер переходить у passive-error, а потім у bus-off. Апаратна логіка включає мультиплексори, які переводять контакти CAN_H/CAN_L на резервний канал, одночасно випускаючи сигнал у центральний шлюз ECU та SIEM/SOC. FIFO-буфери гарантують, що жоден кадр не загубиться під час перемикання, а система

верифікує успішне встановлення зв’язку на резервному каналі перед тим, як припинити моніторинг основного й позначити його як відновлений лише після успішного тесту цілісності. Цей процес відбувається прозоро для вищих рівнів контролю, зберігаючи затримки у межах тимчасових вимог до передачі критичних команд (у деяких випадках до 1-2 мс).

Окрему увагу приділяють розробці та впровадженню безпечних станів (fail-safe modes), які активуються при неможливості забезпечити достатній рівень зв’язку. Згідно зі стандартами ISO 26262 та SAE J2980, для кожної критичної функції має бути чітко визначений Safe State та алгоритм повернення в нормальний режим. Наприклад, якщо зв’язок з ABS ECU втрачається, система переходить у механічний режим гідроприводу гальм, відключаючи електронне регулювання тиску, та виводить на приладову панель сигнали тривоги й рекомендацію зупинитися. Для EPS Safe State полягає в зменшенні або відключенні підсилювача керма, що переводить транспортний засіб у ручний режим із більшим зусиллям на кермо. У разі відмови датчика положення дросельної заслінки двигун переводиться в limp-home режим із обмеженням максимальної швидкості та оборотів, що дозволяє доїхати до сервісу.

У комплексі фізичне дублювання шин, power-line резервування, логічне дублювання ECU, швидкі апаратні алгоритми перемикання та ретельно прописані безпечні стани утворюють багаторівневий захист.

2.5. Логування

Логування в CAN-шині є критично важливим процесом для забезпечення моніторингу, діагностики та підтримки безпеки в сучасних автомобільних і промислових системах. Оскільки сама CAN-шина спроектована як швидкий і надійний протокол обміну повідомленнями між електронними блоками управління (ECU), вона не містить вбудованих механізмів шифрування або автентифікації. За відсутності таких базових засобів захисту можливі спроби підміни даних, атаки типу відмова в обслуговуванні та перехоплення

повідомлень. Саме тому правильна організація процесу логування – від фіксації всіх активностей на шині до надійного збереження й наступного аналізу є однією з головних ланок у побудові безпечної та стійкої інфраструктури на основі CAN.

Першочерговою метою логування є збір максимально повної інформації про трафік на шині: кожне передане або прийняте повідомлення супроводжується записом ідентифікатора, даних корисного навантаження, часової мітки та служби помилок (як-от кадри з бітами помилок або відмов). Така деталізація дозволяє інженерам у разі потреби відтворити події й провести форензичний аналіз (forensic analysis), швидко ідентифікувати аномальні патерни поведінки системи та знайти вузькі місця в роботі ECU. Крім безпеки, логи допомагають оптимізувати архітектуру мережі, коригувати налаштування фільтрації кадрів та підвищувати продуктивність, адже аналіз статистики передаваних повідомлень дає чітке уявлення про характер навантаження та затримки.

Найбільш популярним і надійним підходом до збору даних із CAN-шини є використання апаратних логерів. Ці прилади підключаються безпосередньо до шинного інтерфейсу й автономно записують трафік на внутрішню або змінну пам’ять без суттєвого навантаження на саму мережу. Наприклад, пристрої серії CANedge від CSS Electronics (Рис. 2.5) здатні вести запис повідомлень із частотою до кількох тисяч кадрів за секунду та зберігати їх на SD-картку об’ємом від восьми до тридцяти двох гігабайт. Окрім канальних даних, такі логери зазвичай оснащені вбудованими модулями GPS/IMU, що дозволяє зв’язати кожну подію в мережі з географічними координатами та інформацією про прискорення або орієнтацію пристрою. Це особливо корисно в галузях телематики та віддаленого моніторингу автотранспорту.

Рисунок 2.5 – CANedge від CSS Electronics

При виборі апаратного логера також варто звернути увагу на вбудовані функції фільтрації та тригерів. Так, модель Kvaser Memorator 2xHS v2 (Рис. 2.6) одночасно моніторить два незалежні високошвидкісні канали й дозволяє заздалегідь визначити умови, за яких починається запис (наприклад, поява повідомлення з конкретним ідентифікатором або фреймом з помилкою). Це дає змогу значно зменшити обсяг записаних даних та сфокусуватися на критичних подіях. Існують також рішення, як-от REXGEN Pro від Influx Technology (Рис. 2.7), розроблені для експлуатації в умовах підвищених вібрацій і температурних коливань, які підтримують одночасний логінг до чотирьох каналів CAN FD, а також сумісні з протоколами LIN, LTE і GNSS для передачі зібраних даних у хмару.

Рисунок 2.6 – Kvaser Memorator 2xHS v2

Рисунок 2.7 – REXGEN Pro від Influx Technology

Окрім апаратних засобів, існують і програмні інструменти для логування, які зазвичай використовуються в лабораторіях, на виробництві або під час розробки прошивок ECU. Інструмент CANalyzer від Vector Informatik (Рис. 2.8), наприклад, дозволяє в реальному часі приймати й відправляти CAN-повідомлення, відстежувати стан шинних ліній, виконувати складні сценарії тестування й зберігати результати у зручному форматі. Безкоштовний софт CanKing від Kvaser, навпаки, пропонує базовий функціонал для відладки: фільтрацію повідомлень, емуляцію помилок та логування з використанням тригерів. Відкрите програмне забезпечення BUSMASTER (Рис. 2.9) забезпечує можливість не лише запису трафіку, а й моделювання віртуальних вузлів та тестування поведінки системи під різним навантаженням, що є безцінним при розробці й перевірці алгоритмів обробки CAN-даних.

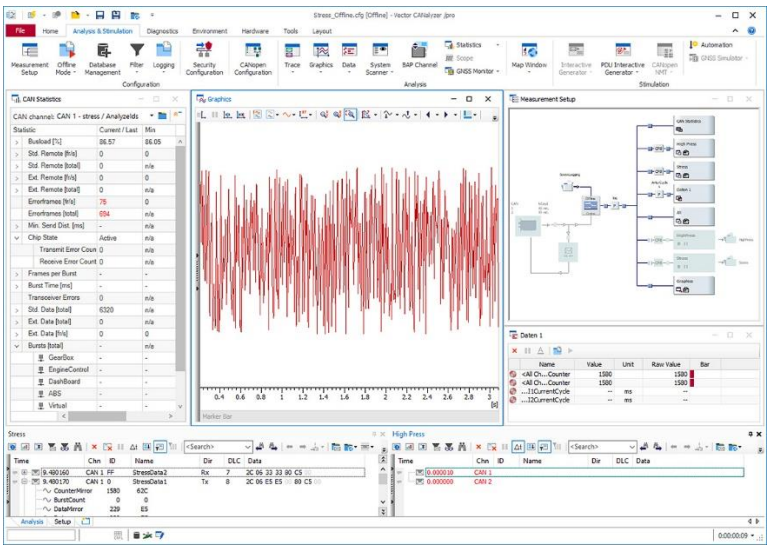

Рисунок 2.8 – CANalyzer від Vector Informatik

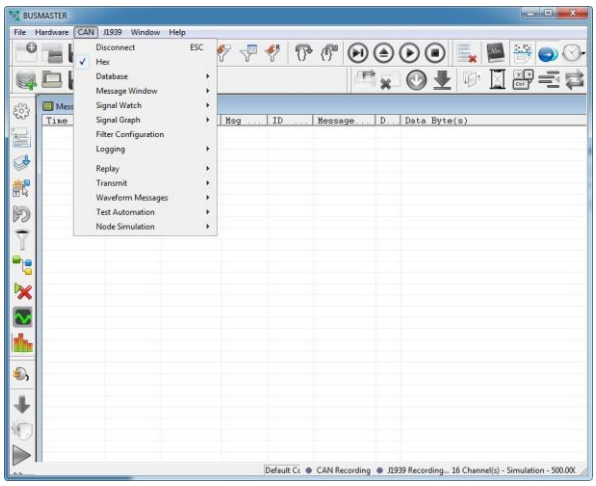

Рисунок 2.9 – BUSMASTER

Ще одним способом організації логування є використання шлюзів та адаптерів, які поєднують CAN-шину з іншими інтерфейсами або мережами. Прикладом є модуль CANmod.router від CSS Electronics (Рис. 2.10), який дозволяє об’єднати до чотирьох шин у єдину логічну мережу та передавати всі дані через інтерфейс USB або Ethernet. Така архітектура зручна для централізованого моніторингу великого парку обладнання – замість підключення окремого логера до кожної шини достатньо встановити шлюзи в ключові точки мережі та відправляти пакетований трафік на центральний сервер. Аналогічно, модуль CANmod.input дозволяє перетворювати на шинні повідомлення аналогові, цифрові й імпульсні сигнали з зовнішніх сенсорів, що дає змогу підключати до CAN-мережі різноманітні датчики без розробки додаткового апаратного забезпечення.

Рисунок 2.10 – CANmod.router від CSS Electronics

При виборі засобів логування варто брати до уваги низку критеріїв. По-перше, тип системи: для віддалених автологістичних застосунків і полів випробувань потрібні автономні апаратні логери з довгим терміном роботи від батарей і високою захищеністю корпусу, тоді як у лабораторних і виробничих середовищах достатньо програмних рішень, що дозволяють швидко налаштовувати сценарії тестування. По-друге, обсяг даних: при частоті обміну тисячі кадрів за секунду й більш важливо мати фільтрацію на апаратному рівні або поділ логів на сегменти, аби уникнути перевантаження пам’яті та каналів зв’язку. По-третє, вимоги до інтеграції: якщо логи мають передаватися в системи SOC/SIEM або хмарні платформи, потрібно обирати пристрої з підтримкою відповідних протоколів зв’язку (TCP/IP, MQTT, HTTPS) і можливістю шифрування даних. І, нарешті, бюджет: інвестиція у високоточні апаратні логери чи корпоративні ліцензії на програмне забезпечення може бути значною, тож важливо зіставити потреби проєкту з фінансовими ресурсами.

Після того як рішення з логування обране і налаштоване, необхідно розробити політику зберігання та обробки зібраних даних. Рекомендується впроваджувати стратегії ротації логів, архівації й резервного копіювання, а також застосовувати хешування чи цифрові підписи для забезпечення цілісності записів. Використання стандартних форматів файлів (наприклад, .asc або .blf для апаратних логерів, .csv чи .json для програмних) гарантує сумісність із численними інструментами аналізу.

Висновки за розділом 2

У підсумку, розглянуті методи захисту CAN-шини демонструють, що надійна безпека мережі досяжна лише за умови багаторівневого підходу. Фізичне сегментування та ізоляція шини створюють першу лінію оборони, обмежуючи поширення потенційних атак та зменшуючи вплив компрометації одного сегмента на всю систему. Використання CAN-фаєрволів і протокольних шлюзів

додає можливість контролю та фільтрації повідомлень на апаратному рівні, перехоплюючи й блокуючи шкідливі фрейми ще до їх обробки ECU.

Системи виявлення аномалій, як фізичні, так і поведінкові (Physical та Behavioral IDS), забезпечують постійний моніторинг мережевого трафіку з метою виявлення нетипових патернів і автоматичного сповіщення про підозрілі зміни в обміні даними. Завдяки їм можна оперативно реагувати на емпіричні ознаки атак, які не завжди підпадають під класичні правила фільтрації. Резервування каналів і впровадження безпечних станів забезпечують стійкість системи до відмов і навмисних спроб перевантаження шини, підтримуючи критичні функції навіть за умови виникнення помилок або атак.

Нарешті, логування подій на CAN-шині створює основу для форензічного аналізу та історичного огляду інцидентів. Комплексне застосування всіх п’яти підходів – від сегментації та фільтрації до детального моніторингу й архівації логів, формує ефективну систему захисту, яка враховує як технічні, так і організаційні аспекти безпеки CAN-мереж.

РОЗДІЛ 3
ЗАСОБИ ЗАХИСТУ CAN-ШИНИ

3.1. Огляд комерційних апаратних міжмережевих екранів (CAN-Firewall)

3.1.1. IXXAT CAN@net NT 420

IXXAT CAN@net NT 420 – це високопродуктивний апаратний модуль, що розроблений компанією HMS Industrial Networks для інтеграції та захисту CAN- і CAN FD-мереж (Рис. 3.1). Завдяки своїй архітектурі пристрій поєднує в собі функціонал шлюзу, моста та інтерфейсу для ПК, що дає змогу організувати безпечний обмін даними між кількома сегментами шини й Ethernet-мережею. У ролі шлюзу CAN@net NT 420 дозволяє здійснювати доступ до CAN-мережі через TCP/IP-сокет на базі простого ASCII-протоколу. Це забезпечує прозору передачу повідомлень із мінімальними затримками та сумісність із широким спектром операційних систем. Функція моста дає можливість об’єднувати розрізнені CAN-сегменти з різними швидкостями передачі, а інтерфейс для ПК через драйвер VCI відкриває доступ до стандартних засобів діагностики, таких як canAnalyser, а також дозволяє розробляти власні аналітичні та тестові додатки.

Рисунок 3.1 – IXXAT CAN@net NT 420 від HMS Industrial Networks

Пристрій обладнано чотирма незалежними каналами CAN, два з яких можуть працювати в режимі CAN FD, що істотно підвищує пропускну здатність мережі – арбітражна швидкість класичного CAN досягає 1 Мбіт/с, а режим передачі даних CAN FD – до 8 Мбіт/с. Вбудована система фільтрації повідомлень дозволяє зменшити зайве навантаження на шину та захистити критично важливі елементи від небажаних чи шкідливих кадрів, відбираючи дані за ідентифікаторами й вмістом полів. У разі необхідності пристрій може автоматично транслювати ідентифікатори, мультиплексувати дані і виконувати складні правила маршрутизації між сегментами з різними специфікаціями та форматами кадрів.

Важливим аспектом є апаратна надійність пристрою. Гальванічна ізоляція відокремлює CAN-інтерфейси від живлення й Ethernet-ліній, що убезпечує від електричних перешкод та перенапруг. Блок захисту від коротких замикань і перенапруг гарантує стабільну роботу навіть у складних промислових чи автомобільних умовах, де можливі стрибки напруги або перешкоди від навколишнього обладнання. Корпус із захистом IP20 дозволяє експлуатувати модуль всередині захищених шаф та панелей.

У виробничих і автомобільних системах застосування CAN@net NT 420 забезпечує сегментацію мережі, що важливо для ізоляції критичних ECU від діагностичних інтерфейсів та потенційно небезпечних вузлів. Це дозволяє запобігати несанкціонованому доступу через OBD-порти або бездротові підключення та знижує ризик DoS-атак, спричинених перевантаженням шини. Виконуючи функцію міжмережевого екрану, пристрій істотно підвищує загальну стійкість системи до зовнішніх загроз, одночасно зберігаючи високу швидкість обміну даними та мінімальні затримки.

З-поміж інших переваг CAN@net NT 420 варто відзначити простоту налаштування через інтуїтивно зрозумілий інтерфейс, масштабованість мережевої топології та можливість гнучкої адаптації під різні сценарії – від

автомобілів і вантажівок до промислових автоматизованих ліній, будівельної автоматизації й медичного обладнання.

3.1.2. PlaxidityX CAN Protection

PlaxidityX CAN Protection – це комплексне рішення для виявлення та запобігання несанкціонованим втручанням у CAN-мережі, розроблене компанією PlaxidityX (Рис. 3.2). У своїй роботі система аналізує трафік у реальному часі, порівнюючи потоки даних із завантаженими профілями допустимої поведінки та виявляючи аномалії, що можуть сигналізувати про атаки типу CAN injection або Denial-of-Service. Завдяки цьому PlaxidityX CAN Protection дозволяє вчасно реагувати на підозрілі дії: воно не тільки фіксує спроби зловмисного впливу на шину, а й блокує небезпечні пакети, мінімізуючи ризики збоїв у роботі критичних підсистем автомобіля – від іммобілайзера до центрального замка.

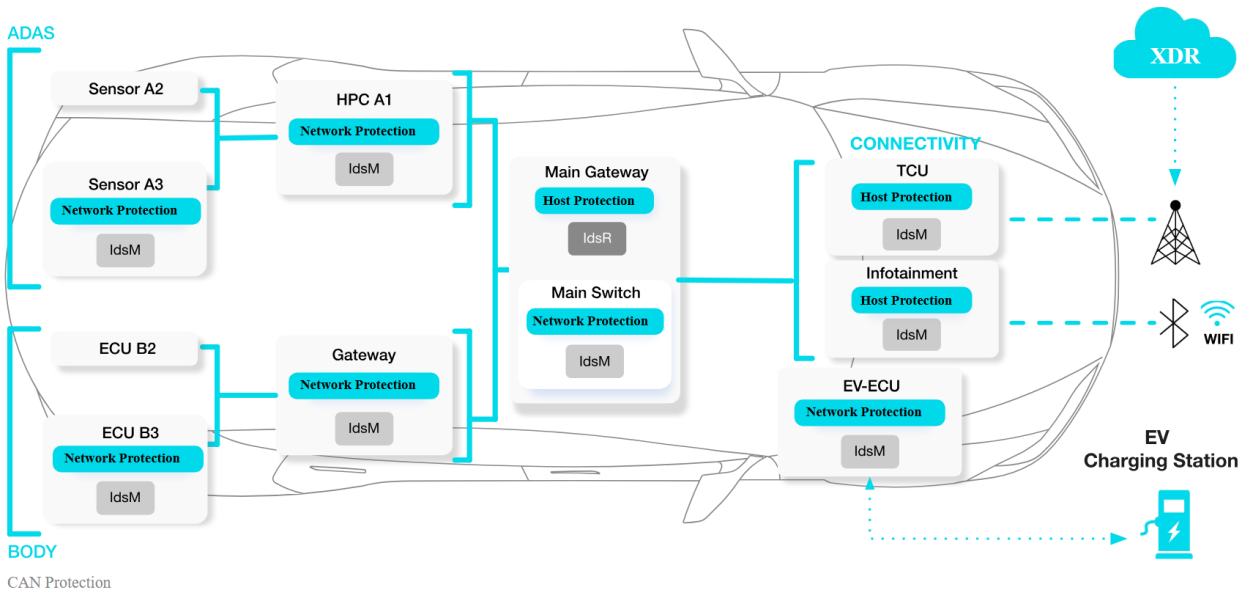

Рисунок 3.2 – Схема PlaxidityX CAN Protection

Особливістю рішення є глибока інтеграція зі стандартом AUTOSAR через підтримку EB tresos, що забезпечує злагоджену взаємодію з іншими

компонентами програмного забезпечення електронних блоків керування (ECU). Така сумісність спрощує розгортання PlaxidityX CAN Protection на виробництві та у процесі обслуговування автопарку, а також гарантує відповідність вимогам, пред'явленим провідними автовиробниками. Крім того, оптимізована архітектура системи мінімізує навантаження на обчислювальні ресурси ECU, що дозволяє зберегти високу продуктивність мережі без шкоди для безпеки. Навантаження на процесор лишається незначним навіть під час інтенсивного обміну повідомленнями, що характерно для сучасних автомобільних систем з великою кількістю сенсорів і виконавчих механізмів.

PlaxidityX CAN Protection вирізняється гнучкістю настроювання: завдяки інструментам конфігурації можна адаптувати політики безпеки під будь-яку архітектуру транспортного засобу – від легкових автомобілів до вантажівок і комерційних автобусів. Використовуючи шаблони профілів або створюючи власні правила фільтрації та реагування, інженери можуть встановити різні рівні жорсткості для кожного сегмента шини: додатковий контроль діагностичних портів, посилений моніторинг важливих ECU або лише пасивне оповіщення про аномалії. Така масштабованість робить рішення універсальним для побудови захищених мереж як у легкових, так і в промислових та медичних застосуваннях, де CAN-інтерфейси використовуються для управління критичними процесами.

3.1.3. NXP S32K3 Security Gateway Module

NXP S32K3 Security Gateway Module – це рішення для захисту та інтеграції автомобільних мереж, розроблене компанією NXP Semiconductors на основі мікроконтролерів серії S32K3 з ядрами Arm® Cortex®-M7 (Рис. 3.3). Завдяки одно- та двоядерним конфігураціям, а також lockstep-режиму, модуль забезпечує обробку складних алгоритмів кібербезпеки та маршрутизації даних із тактовою частотою до 320 МГц. Вбудований апаратний модуль безпеки (HSE) підтримує провідні криптографічні алгоритми AES, RSA та ECC, що дає змогу організовувати зашифровані канали обміну в мережах CAN FD.

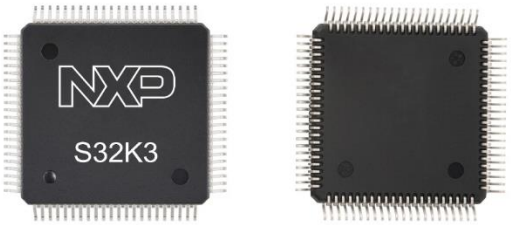

Рисунок 3.3 – S32K3 Microcontroller

Конструкція Security Gateway Module (Рис. 3.4) відповідає вимогам ISO 26262 до рівня ASIL D, що гарантує функціональну безпеку критичних систем. Пристрій підтримує до шести портів CAN FD і чотири порти LIN, а також Ethernet-інтерфейси з підтримкою Time-Sensitive Networking (TSN) і Audio Video Bridging (AVB), що дозволяє реалізувати складні багаторівневі топології з низькими затримками. Вбудована підтримка OTA-оновлень (FOTA) забезпечує безпечне дистанційне оновлення прошивки, мінімізуючи ризики та операційні витрати на технічне обслуговування.

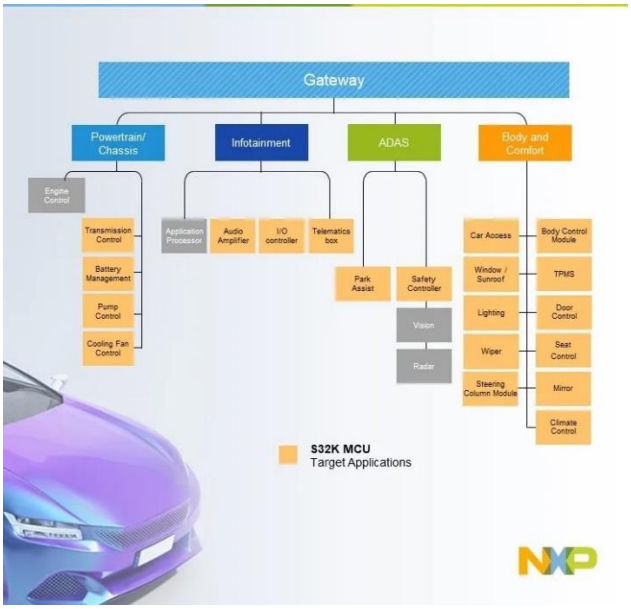

Рисунок 3.4 – NXP S32K3 Security Gateway Module

Серед ключових переваг цього модулю – висока обчислювальна потужність і масштабованість: різні конфігурації ядер і кількості інтерфейсів дозволяють

адаптувати рішення під різні архітектури та вимоги замовника. Підтримка сучасних стандартів безпеки та телематики робить S32K3 Security Gateway хорошим вибором для побудови автомобільних мереж – від легкових авто та комерційного транспорту до автономних та інтелектуальних транспортних систем.

3.2. Системи виявлення та попередження вторгнень (IDS/IPS) для CAN

3.2.1. Vector CANoe

Створена компанією Vector Informatik, ця програмна система призначена для роботи з найактуальнішими стандартами, такими як CAN, CAN FD, LIN, FlexRay, MOST, Ethernet та з протоколами вищого рівня – J1939, CANopen, ARINC 825 та ін. Основною метою Vector CANoe (Рис. 3.5) є забезпечення інженерів інструментами для побудови віртуальних моделей ECU, симуляції мережевих топологій і конфігурації сценаріїв обміну даними, що робить її надзвичайно корисною на різних етапах життєвого циклу програмно-апаратних комплексів автомобільної електроніки.

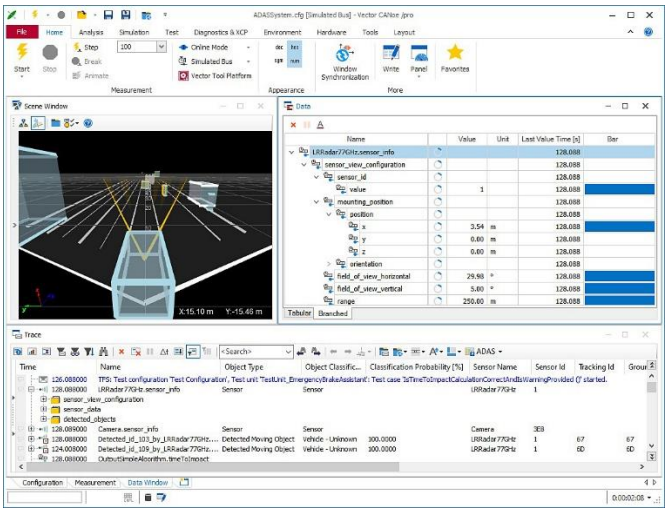

Рисунок 3.5 – Vector CANoe

Серед багатьох переваг цього інструменту – можливість створення топології мережі в графічному інтерфейсі. Інженери можуть описати фізичні та логічні з’єднання між ECU, встановити швидкість обміну для кожного сегмента шини та задати правила арбітражу повідомлень. Після побудови моделі мережі CANoe дозволяє підключати як реальні апаратні модулі, так і віртуальні імітації ECU, що дає змогу проводити гібридні тести типу Hardware-in-the-Loop (HIL). У такому середовищі розробники здатні перевірити взаємодію справжніх контролерів з моделями інших вузлів без необхідності будувати повноцінний фізичний прототип транспортного засобу.

Для симуляції логіки роботи ECU CANoe використовує власну мову програмування CAPL (Communication Access Programming Language). Завдяки CAPL-скриптам можна реалізувати будь-яку алгоритмічну поведінку: від простої генерації тестових повідомлень до складних сценаріїв обміну з різним часовим зрушенням і логічними умовами. CAPL-скрипти інтегруються безпосередньо в проєкт CANoe, що дозволяє створювати динамічні тести та реагувати на події в мережі в режимі реального часу. Це забезпечує високий ступінь автоматизації тестування, скорочуючи час і зусилля на налаштування кожного окремого сценарію.

У контексті кібербезпеки Vector CANoe є інструментом для розробки та верифікації систем виявлення вторгнень (IDS) у CAN- та CAN FD-мережах. Інженери можуть спеціально моделювати атаки типу CAN injection, імітувати безперервну передачу зловмисних кадрів або створювати ситуації Denial-of-Service шляхом відправлення повідомлень із високою частотою. Далі ці сценарії аналізуються за допомогою логічних правил і CAPL-скриптів, які імітують поведінку IDS, наприклад, перевірку часових інтервалів між кадрами або вмісту даних. Отримані результати дозволяють визначити, наскільки ефективно розроблені механізми безпеки здатні виявляти й блокувати підозрілі пакети.

Ключовим аспектом CANoe є інтеграція з MATLAB і Simulink, що відкриває додаткові можливості для інженерів із моделювання систем керування. Моделі, створені в Simulink, можна імпортувати до CANoe і запускати в

середовищі HIL, перевіряючи їхню поведінку за допомогою реальних шинних інтерфейсів або віртуальних ECU. Це забезпечує єдину платформу для розробки алгоритмів керування, їхнього тестування та калібрування в реальних умовах. Така інтеграція істотно прискорює час виходу продукту на ринок і підвищує якість кінцевих рішень.

Ще одним потужним інструментом у складі CANoe є опції аналізу часу реального роботи мережі. На основі даних про затримки доставки кадрів, втрати повідомлень і зміни часових інтервалів інженери можуть будувати гістограми, графіки часового ряду й оцінювати відповідність параметрів системи вимогам до реального часу. Особливо важливо це для мереж, в яких використовуються критичні протоколи, що вимагають гарантованої доставки даних із певним часовим обмеженням. Завдяки таким інструментам розробники отримують чітке розуміння продуктивності мережі та можуть оперативно коригувати налаштування шинних інтерфейсів і параметри ECU.

3.2.2. ESCRYPT CycurIDS

Система ESCRYPT CycurIDS (Рис. 3.6) була створена як відповідь на зростаючі вимоги до кібербезпеки сучасних транспортних засобів, які дедалі більше інтегрують електронні блоки управління та мережеві інтерфейси для обміну даними. У базі рішення лежить модульна архітектура, що дозволяє гнучко масштабувати його функціональність – від простого моніторингу CAN-шин до комплексного захисту багаторівневих мереж, що включають Ethernet, LIN та інші шинні технології. Кожен із компонентів CycurIDS виконує свою чітко визначену роль: датчики встановлюються безпосередньо в ECU та збирають телеметрію, менеджер безпеки об’єднує події, а модуль звітування гарантує своєчасну передачу інформації до центру операцій безпеки транспортного засобу.

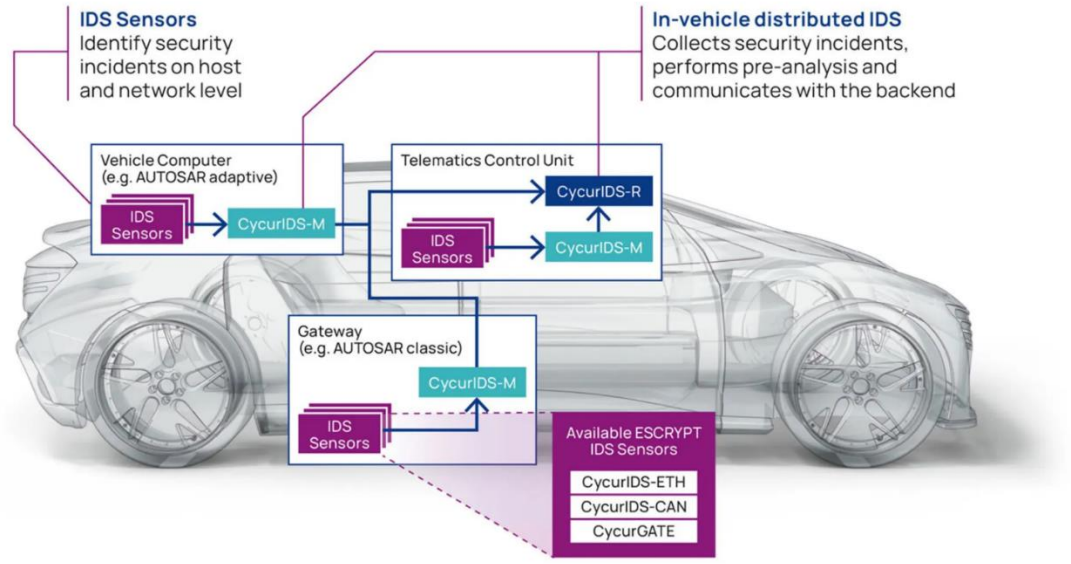

Рисунок 3.6 – Система ESCRYPT CycurIDS

На рівні контролю CAN-трафіку ключовим елементом є модуль CycurIDS-CAN, який вбудовується безпосередньо в блок управління або виконується на виділеному безпековому ECU. Він постійно аналізує потоки кадрів, порівнюючи їх із сформованими профілями поведінки, що будуються на основі специфікацій мережі (файли DBC або ARXML). Завдяки цьому рішення виявляє аномалії у форматі та частоті повідомлень, включаючи випадки CAN injection, спроби несанкціонованої діагностики та DoS-атаки шляхом перевантаження шини. Визначивши підозрілі відхилення, CycurIDS-CAN може не лише зафіксувати подію, але й ініціювати локальну дію – наприклад, блокувати потік кадрів або сигналізувати іншим компонентам системи про загрозу.

Паралельно з цим для Ethernet-мереж передбачений модуль CycurIDS-ETH, який використовує сигнатурний аналіз та машинне навчання для виявлення нетипових запитів під час обміну телеметрією і керувальними командами. Оскільки сучасні автомобілі все частіше застосовують протоколи IPv4/IPv6, MQTT та HTTP(S) для взаємодії з хмарними платформами та мобільними додатками, важливо виявляти спроби несанкціонованої зміни конфігурацій, перенаправлення запитів або підміни даних. Модуль CycurIDS-ETH аналізує як заголовки пакетів, так і їхній вміст, порівнюючи з еталонними профілями та

виявляючи невідповідності, що можуть свідчити про атаку на транспортний маршрут або OBD-інтерфейс.

Усі зібрані події маршрутизуються до менеджера безпеки CycurIDS-M, який виступає єдиною точкою агрегації інформації про інциденти. Цей компонент об’єднує сигнали від усіх сенсорів і виконує кореляцію подій за часом і джерелом, що дозволяє створити повну картину атаки та визначити ланцюжок її етапів. CycurIDS-M підтримує стандартизовані інтерфейси AUTOSAR Classic та Adaptive, що спрощує інтеграцію у вже наявну архітектуру ECU, і забезпечує високу швидкість обробки великих обсягів подій у реальному часі.

Однією з найважливіших переваг ESCRYPT CycurIDS є його відповідність нормативним вимогам UN R155 та ISO/SAE 21434. Рішення розроблене з урахуванням фаз життєвого циклу автомобіля – від проектування та виробництва до експлуатації та утилізації – що гарантує послідовне застосування процедур оцінки ризиків, управління під час впровадження та постійного моніторингу під час експлуатації. При цьому виробник може автоматизувати процес формування правил детекції на основі опису мережі та моделей поведінки, що значно скорочує час на налаштування та знижує ймовірність помилок.

Технічна інтеграція CycurIDS передбачає використання апаратних платформ, сертифікованих на відповідність рівням ASIL B–D за ISO 26262, що дозволяє розгортати рішення навіть у висококритичних системах – наприклад, у блоках керування гальмами чи підвіскою. Крім того, завдяки партнерству з провідними виробниками мікроконтролерів та ECU, ESCRYPT оптимізував свої модулі під специфічні апаратні ресурси, забезпечивши високу продуктивність аналізу під час мінімального навантаження на процесор та оперативну пам’ять.

У практичних проектах застосування CycurIDS продемонстрував високу ефективність: у ході польових випробувань системи великого автопарку зафіксовано та відреаговано на понад 200 інцидентів, серед яких були спроби ін’єкцій некоректних CAN-повідомлень та DoS-атаки.

3.3.3. AUTOSAR SAE J1939 Security Extensions

Автомобільні мережі на базі протоколу SAE J1939 стали галузевим стандартом для комунікації між електронними блоками управління (ECU) у комерційній техніці – вантажівках, автобусах, сільськогосподарських машинах і спеціальній будівельній техніці. Протокол J1939, побудований поверх класичного CAN, гарантує надійну передачу даних у реальному часі та забезпечує багатий набір параметричних групових номерів (PGN) для розподілу повідомлень за функціональними категоріями. Однак з розвитком концепції “connected car” і появою нових інтерфейсів для віддаленої діагностики, оновлення ПЗ та телеметрії в хмару, у мережі J1939 виникли критичні вразливості. На відміну від Ethernet-рішень, CAN та J1939 не передбачають вбудованих механізмів автентифікації, цілісності або конфіденційності повідомлень. Відповідно, без додаткових засобів захисту шина залишається відкритою до атак типу man-in-the-middle, ін’єкцій шкідливих кадрів, повторного відтворення повідомлень (replay attacks) та відмови в обслуговуванні (DoS).

У відповідь на нові виклики AUTOSAR (відкритий консорціум виробників автомобільної електроніки), розробив модуль Secure Onboard Communication (SecOC), який інтегрує в екосистему AUTOSAR стандартизовані механізми автентифікації та захисту цілісності повідомлень. Основною ідеєю SecOC є додавання до кожного фрейма CAN криптографічного коду автентифікації повідомлень (Message Authentication Code, MAC), а також лічильника свіжості (Freshness Counter), який унеможливлює повторне відтворення раніше зафіксованих повідомлень. Таке рішення дозволяє отримувачу перевірити, що кадр походить від справжнього відправника і не був змінений у дорозі, а також гарантує, що повідомлення є абсолютно новим, а не реплейованим.

У традиційному SecOC для CAN передбачається, що відправник інкрементує свій внутрішній лічильник за кожним відправленим MAC-закріпленим кадром, а отримувач зберігає останнє значення і вимагає, щоб кожне нове повідомлення мало лічильник вище за попереднє. У J1939 можливі

затримки або втрата кадрів через архітектуру мультикасту, тому при адаптації передбачаються діапазони дозволених значень і механізми ресинхронізації лічильника у разі значних стрибків, щоб уникнути хибних спрацьовувань.

На рівні специфікації AUTOSAR Secure Onboard Communication для J1939 також прописані процедури управління ключами криптографії. Використання симетричних алгоритмів спрощує завдання для ресурсів ECU. AUTOSAR передбачає застосування централізованого генератора ключів (Key Management Service) та міжмодульний інтерфейс для оновлення ключів у полі (реалізація FOTA-процедур). При цьому ключі можуть бути індивідуальними для кожного ECU або груповими для підсистем, що обмінюються повідомленнями в межах визначеного PGN-діапазону.

Упровадження розширень безпеки AUTOSAR для J1939 на практиці стикається з низкою викликів. По-перше, термін життя комерційних транспортних засобів зазвичай перевищує 10–15 років, а оновлення ECU дорого обходиться. Тому нові механізми повинні бути сумісні зі старими блоками, які можуть не мати апаратної підтримки SHA-256, AES-дескриптора чи достатнього обсягу пам’яті для зберігання MAC і лічильників. По-друге, виробники різняться за підходами до хмарної телеметрії та діагностики, тому модель розповсюдження ключів та політик може відрізнятися залежно від екосистеми. Нарешті, обмеження ресурсів можуть накладати обмеження на довжину повідомлень і частоту передачі, тому SecOC під J1939 необхідно оптимізувати для мінімального зростання затримок.

Незважаючи на складнощі, результати раннього впровадження показують суттєве підвищення стійкості мережі до атак. У пілотних проєктах, проведених спільно з кількома великими операторськими компаніями, відбулося зниження кількості успішних ін’єкцій шкідливих кадрів на понад 90 %, а технології oto-on-the-fly оновлення ключів дозволили впроваджувати поліпшення безпеки під час планових технічних обслуговувань без витрати додаткового часу простою. Крім того, стандартизація процедур ресинхронізації та обробки пропущених кадрів

дозволила уникнути хибних спрацьовувань у понад 95 % сценаріїв, навіть у мережах із пропускною спроможністю нижче 250 кбіт/с.

3.4. Open-source інструменти та акселератори розробки

3.4.1. CANtact + can-utils (SocketCAN)

У сучасному світі, де автомобільні та промислові системи дедалі більше автоматизуються та об’єднуються в єдині екосистеми, вивчення та тестування протоколу CAN набуває особливої авжливості. Не кожна лабораторія або невелика компанія може дозволити собі дорогі комерційні аналізатори та емулятори, саме тому інструменти з відкритим кодом стають ключовим ресурсом для інженерів, дослідників і студентів. Однією з найуспішніших спроб поєднати доступність апаратної платформи з потужним і зрозумілим програмним інтерфейсом є проєкт CANtact (Рис. 3.7) в поєднанні зі стеком SocketCAN (Рис. 3.8) та набором утиліт can-utils.

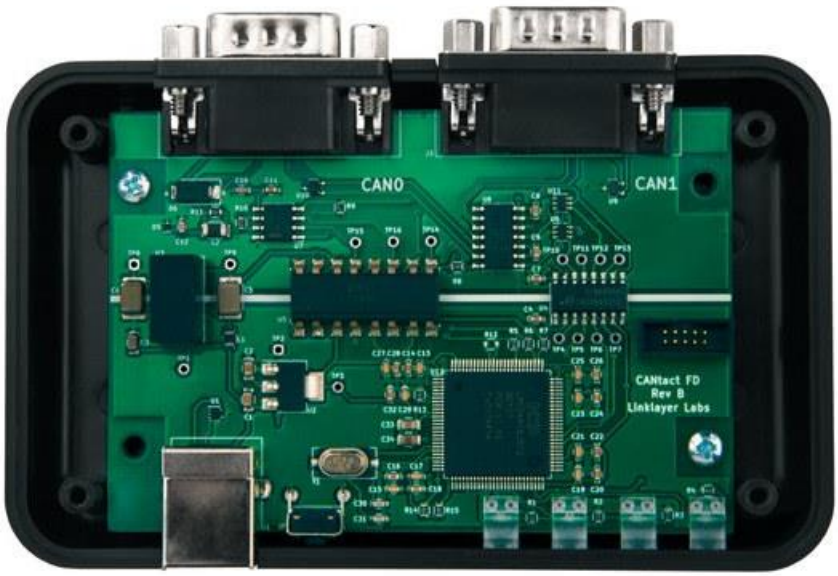

Рисунок 3.7 – CANtact

CANtact є апаратним пристроєм USB-to-CAN, розробленим із дотриманням принципів відкритого дизайну: всі схеми, компоненти та програмна частина доступні у відкритому репозиторії. Пристрій підключається до ПК через стандартний USB-інтерфейс, а до шини через роз'єм DB9 з підтримкою класичного CAN та розширених режимів CAN FD у версії Pro. Відмінною рисою CANtact є простота конструкції та зручність у використанні: він не потребує складного налаштування драйверів – в Linux-системах його підтримує нативний стек SocketCAN, у Windows можна використовувати вільні альтернативні драйвери, а для macOS існують проекти з частковою підтримкою. Завдяки відкритості апаратної платформи кожен користувач може змінити схему живлення, додати ізоляцію або модифікувати прошивку мікроконтролера для власних потреб. Також доступна версія CANtact Pro, де реалізовано два незалежні канали, попередню обробку сигналів і гальванічну ізоляцію USB-шини від CAN-мережі.

SocketCAN – набір драйверів і утиліт у складі ядра Linux, який перетворює CAN-інтерфейси на звичайні мережеві пристрої. Підключивши CANtact, користувач отримує інтерфейс з іменем на кшталт can0 або can1, який можна підняти за допомогою стандартної команди: ip link set can0 up type can bitrate 500000. Далі він стає повноправним мережевим інтерфейсом, до якого можна підключати програми, що працюють з сокетами, аналізувати повідомлення за допомогою tcpdump або налаштовувати маршрутизацію та фільтри. Щоб полегшити повсякденне застосування, автори проєкту створили набір can-utils – набір утиліт командного рядка, який включає такі інструменти, як candump для запису всієї активності шини в реальному часі, cansend для відправки одиночних або серійних фреймів, cangen для синтетичного навантаження і тестування стійкості мережі, canplayer, який дозволяє відтворювати попередньо записані сесії, а також більш спеціалізовані утиліти на кшталт cansniffer для виявлення змін у даних або cangw для налаштування простих шлюзів між двома CAN-мережами.

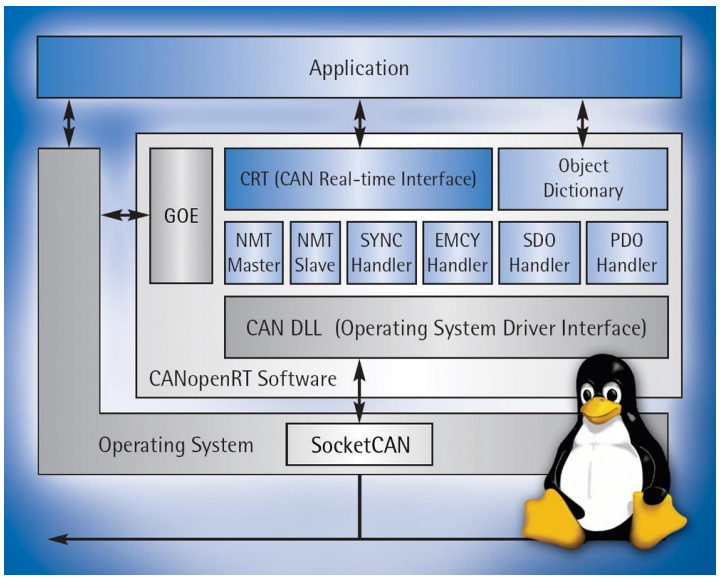

Рисунок 3.8 – SocketCAN

Окрім прямої взаємодії з апаратурою, проєкт відкритий до розширень та інтеграції з іншими інструментами. Наприклад, існують бібліотеки на Python та C/C++, що включають функції can-utils, і дозволяють будувати власні GUI-додатки або автоматизовані тести з гнучким сценарієм. Ком'юніті регулярно ділиться своїми напрацюваннями на GitHub та форумах, обговорюючи методи оптимізації таймінгів, поліпшення фільтрації або додавання підтримки нових стандартів, таких як CAN FD або LIN-over-CAN. Така відкритість сприяє швидкому обміну досвідом і зростанню якості коду: будь-хто може внести свій патч, пройти код-рев'ю та стати офіційним контриб'ютором проєкту.

Практичні кейси застосування CANtact та can-utils включають не лише автомобільні системи. Промислові контролери, роботизовані лінії, медицинські прилади і навіть енергетичні мережі на базі CAN-протоколу успішно діагностуються та налагоджуються за допомогою тих самих наборів інструментів. Виділена ізоляція та підтримка CAN FD у версії Pro дають змогу використовувати CANtact у середовищах із високим рівнем шуму та суворими вимогами до пропускної здатності. Додатково до цього, можливість роботи в багатокористувацькому режимі, коли кілька процесів одночасно читають та пишуть CAN-пакети, дозволяє організовувати як внутрішні «чорні ящики» для запису, так і зовнішні моніторингові системи.

3.4.2. can-isotp / cantools в Python

У середовищі Python для роботи з ISO-TP та CAN створено дві взаємодоповнювані бібліотеки – can-isotp і cantools, що разом утворюють потужний інструментарій для розробників, дослідників та ентузіастів відкритого коду.

Бібліотека can-isotp реалізує весь механізм ISO-TP на рівні користувача, дозволяючи передавати та отримувати довгі повідомлення через CAN-шину. Вона може працювати як поверх стандартного SocketCAN у Linux, так і у власному режимі, керуючись таймінгом і розрахунком блоків без залучення низькорівневих драйверів ядра. Розробники отримують гнучкість налаштування розміру блоку, інтервалів між кадрами та тайм-аутів, що особливо важливо для складних сценаріїв, коли потрібно адаптуватися до обмежень апаратних контролерів або високого рівня шуму в фізичному середовищі. Установка can-isotp здійснюється простою командою pip install can-isotp, після чого можна одразу налаштовувати стек ISO-TP на будь-якому доступному CAN-інтерфейсі.

Паралельно з транспортним рівнем постає завдання інтерпретації та формування вмісту самих повідомлень. Усю цю роботу бере на себе бібліотека cantools, яка читає описові файли мережі (зокрема DBC, KCD, SYM чи ARXML) і на їх основі генерує механізми кодування й декодування сигналів у кадрах CAN. Завантаживши DBC-файл, розробник може створити об’єкт повідомлення, вказавши назви сигналів і їхнє значення, а cantools перетворить це на масив байтів готових для відправки пакетів. Аналогічно, отримавши сирі дані з шини, він декодує їх у набір зрозумілих параметрів, позбавивши інженера необхідності розбирати їх вручну.

Найпоширеніший сценарій використання обох бібліотек виглядає так: на налаштуваній шині формуються потрібні дані за допомогою функцій cantools згідно з DBC-описом. Після цього байти передаються через can-isotp, який пакує їх у низку CAN-фреймів. На приймальній стороні стек збирає всі частини в єдину

послідовність байтів, яку потім знову передають у cantools для декодування в набір названих сигналів і значень.

Ще однією вагомою перевагою використання Python-екстеншенів замість низькорівневих C-бібліотек є можливість легко інтегрувати написані скрипти в більші системи автоматизації та тестування. Наприклад, у середовищі безперервної інтеграції можна організувати цикл автоматичних тестів, який запускає перевірки стійкості транспортних підсистем, генерує сценарії зі зміненими таймінгами, імітує різні режими експлуатації або навіть перевіряє поведінку при втраті кадрів. Поєднання can-isotp та cantools з фреймворками типу pytest або Robot Framework відкриває можливості корпоративної автоматизації, коли кожне оновлення прошивки піддається перевірці протоколу ISO-TP та коректності передачі повідомлень.

Крім чисто технічних аспектів, важливим є фактор спільноти та відкритості коду. Обидві бібліотеки активно підтримуються на GitHub, де користувачі доповнюють документацію, вносять виправлення, додають нові режими адресації або розширюють підтримку нестандартних протоколів, таких як UDS (Unified Diagnostic Services). При цьому документація з прикладами коду, інтегрованими тестами та оглядами реалізації дозволяє новачку швидко освоїтися та почати вносити власні вдосконалення. Широке ком’юніті гарантує не лише швидке усунення помилок, але й постійне оновлення відповідно до нових версій Python, SocketCAN та специфікацій CAN.

У галузі розробки вбудованого програмного забезпечення, де ресурси ECU зазвичай обмежені, інтеграція цих бібліотек допомагає визначити оптимальні параметри шини та програмних алгоритмів перед тим, як остаточно зкомпілювати код для мікроконтролера. Інженери можуть моделювати роботу на своєму комп’ютері, аналізувати затримки, відсоток втрат кадрів при різних налаштуваннях і тільки після цього переносити конфігурацію у прошивку. Такий підхід знижує ризики дорогої ітерації на апаратному стенді й дозволяє виводити готовий до сертифікації продукт більш швидко та надійно.

Практичні кейси включають розробку діагностичних систем, де необхідно передавати значні обсяги даних про стан двигуна або шасі, конфігурувати параметри антиблокувальної системи ABS, оновлювати мікропрограми через OBD-порт або реалізовувати складні взаємодії між кількома типами шин з різними швидкостями.

3.5. Порівняльний аналіз і критерії вибору

3.5.1. Продуктивність (затримка, пропускна здатність)

Ключові показники продуктивності засобів захисту CAN-шини: затримка (час обробки й передачі кадрів) та пропускна здатність (максимальний обсяг даних, що може передаватися за одиницю часу). Вимоги до кожного з цих параметрів різняться залежно від функціональних завдань системи. Автомобільні контролери, зокрема в підсистемах безпеки (ABS, ESP, підсилювач керма), потребують майже миттєвої реакції – затримки не повинні перевищувати декілька мікросекунд. Натомість у діагностичних або інформаційно-розважальних підсистемах допустимі більші затримки. Пропускна здатність CAN класичної версії обмежена 1 Mbit/s, а в CAN FD – до 8 Mbit/s.

Затримка в апаратних міжмережевих екранах (CAN-Firewall)

Апаратні рішення виконують фільтрацію й маршрутизацію кадрів на рівні спеціалізованих ASIC або FPGA, що забезпечує затримки від одиниць до десятків мікросекунд.

- IXXAT CAN@net NT 420 обробляє до 1 000 000 кадрів/с із затримкою ≤5 ms на пакет і підтримує CAN FD до 8 Mbit/s із коефіцієнтом корисного навантаження ≈ 95 %.
- NXP S32K3 Security Gateway Module із вбудованим криптопроцесором HSE демонструє затримки ≤ 10 ms при одночасній роботі

шести портів CAN FD і чотирьох LIN – незалежно від температури (–40…+125 °C) і коливань живлення (12 V ± 20 %).

- PlaxidityX CAN Protection застосовує FPGA-прискорювач для шаблонного аналізу кадрів із додатковою затримкою лише 1-2 ms (≤ 0,1 % від часу доставки).

Затримка в системах виявлення та запобігання вторгнень (IDS/IPS)

Програмні або гібридні механізми аналізують поведінкові характеристики трафіку, що вносить більші затримки, але дозволяє реалізувати глибокий аналіз.

- Vector CANoe у конфігурації Hardware-in-the-Loop обробляє до 1000 кадрів/с із середньою затримкою 2-5 ms у складних сценаріях синхронізації CAN, FlexRay і Ethernet; у більш простих – 1-2 ms.
- ESCRYPT CycurIDS оптимізований для ресурсів ECU: до 100 000 подій/с із затримкою ≤ 1 ms, точність виявлення аномалій ≥ 98 % (хибні спрацьовування ≤ 0,5 %).
- AUTOSAR SecOC додає 1-3 ms для перевірки MAC і лічильника свіжості в PDU-Router, але пряма інтеграція в middleware знижує загальні витрати часу.

Затримка в криптографічних стек-бібліотеках для ECU

Програмні бібліотеки додають затримки обчислень, натомість забезпечують високу гнучкість.

- CANcrypt (AES-CMAC) – 0,5-2 ms затримки на пакет на 32-розрядних MCU без апаратного прискорювача.
- ESCRYPT CycurLIB із апаратним прискоренням HSE у S32K3 – 0,1-0,3 ms на пакет.
- AUTOSAR SAE J1939 SecOC із буферизацією й оптимізованою перевіркою MAC – близько 1 ms затримки при 250 kbit/s.

Таблиця 3.1

	Рішення	**Затримка**	**Пропускна здатність**
CAN-Firewall	IXXAT CAN@net NT 420	≤ 5 ms	≈ 1 Mbit/s (CAN), 8 Mbit/s (FD)
	PlaxidityX CAN Protection	≤ 2 ms	≈ 1 Mbit/s
	NXP S32K3 Security Gateway Module	≤ 10 ms	8 Mbit/s
IDS/IPS	Vector CANoe	2-5 ms	0,8-1 Mbit/s
	ESCRYPT CycurIDS	≤ 1 ms	≈ 1 Mbit/s
	AUTOSAR SecOC	1-3 ms	0,25-1 Mbit/s
Криптографія	CANcrypt	0,5-2 ms	≈ 1 Mbit/s
	ESCRYPT CycurLIB	0,1-0,3 ms	8 Mbit/s
	AUTOSAR SAE J1939 SecOC	≈ 1 ms	0,25 Mbit/s

Апаратні CAN-фаєрволи на основі ASIC/FPGA (IXXAT CAN@net NT 420, NXP S32K3 HSE, PlaxidityX) гарантують мінімальні затримки на рівні одиниць-десятків мікросекунд і пропускну здатність до 8 Mbit/s, що робить їх оптимальними для безпекових систем із жорсткими реальновимірювальними характеристиками. Системи IDS/IPS (Vector CANoe, ESCRYPT CycurIDS, AUTOSAR SecOC) забезпечують гнучкий глибинний аналіз трафіку з затримками у межах 1-5 ms, прийнятними для загального моніторингу, проте менш доцільними в критичних підсистемах реального часу. Криптографічні бібліотеки (CANcrypt, ESCRYPT CycurLIB, AUTOSAR SAE J1939 SecOC) додають від 0,1 до 2 ms затримки на пакет, поєднуючи високий рівень захисту з можливістю масштабування під різні MCU. Таким чином, оптимальне рішення досягається балансуванням між допустимими затримками, пропускною здатністю та рівнем безпеки з урахуванням ресурсів платформи та специфіки застосування.

3.5.2. Ступінь безпеки (криптоалгоритми, сертифікації)

В рамках протоколу SecOC (Secure On-Board Communication) AUTOSAR використовується побудований на базі AES CMAC (Cipher-based Message Authentication Code) для перевірки цілісності повідомлень і підтвердження їхнього походження. Асиметричні схеми ECC, застосовують для первинного встановлення довіри між вузлами мережі або обміну сеансовими ключами, що підвищує загальний рівень захищеності з мінімальними затримками.

Хеші на базі SHA-256 (Secure Hash Algorithm 256-біт) використовуються для формування коротких контрольних суми, які при додаванні до передаваної інформації дозволяють отримувачу перевірити, чи не відбувалося випадкове або навмисне спотворення даних під час транспортування. У низці програмно-апаратних рішень як ESCRYPT CycurLIB передбачена підтримка апаратного прискорення обчислення хеш-функцій та MAC (Message Authentication Code), що дозволяє знизити затримки до субмілісекундних значень навіть на бюджетних мікроконтролерах сімейства 32-розрядних MCU.

Участь засобів захисту CAN-шини в екосистемі автомобільного стандарту ISO/SAE 21434 визначає вимоги до процесів проєктування, розробки, тестування та експлуатації криптографічних механізмів. Стандарт ISO/SAE 21434 охоплює весь життєвий цикл автомобільної електроніки, починаючи з аналізу загроз і закінчуючи виведенням із експлуатації, встановлюючи правила валідації програмного та апаратного забезпечення на відповідність вимогам кібербезпеки. Паралельно застосовується ISO 26262, що фокусується на функціональній безпеці і визначає категорії ASIL (Automotive Safety Integrity Level). Засоби захисту, які інтегруються в підсистеми високого рівня ASIL C або D, повинні проходити додаткові етапи оцінювання впливу криптографічних компонентів на безпеку функцій.

Наявність незалежних сертифікаційних процедур від органів на кшталт TÜV SÜD, UL (Underwriters Laboratories) або Common Criteria (ISO/IEC 15408) підтверджує, що засоби захисту пройшли тестування на витривалість до атак за

заданими сценаріями та відповідають вимогам певного рівня EAL (Evaluation Assurance Level). Наприклад, отримання сертифікації FIPS 140-2 (Federal Information Processing Standard) на криптографічні модулі засвідчує, що використані алгоритми та їхнє апаратне або програмне втілення відповідають суворим вимогам міністерства торгівлі США. Подібні валідації особливо важливі для рішень, які застосовуються в системах допомоги водієві (ADAS) або автономного керування, де ризик кібератак загрожує не лише втраті даних, а й безпеці руху.

У більшості сучасних міжмережевих шлюзів та IDS/IPS рішень (наприклад, Vector CANoe Security Module, ESCRYPT CycurIDS) впроваджуються механізми динамічної ротації ключів, що дозволяє мінімізувати ймовірність компрометації через тривалий термін експлуатації одного й того самого ключа. Зазвичай інтервали ротації визначаються на основі оцінки ймовірності успішної атаки, її вартості та критичності оброблюваних даних. Використання протоколів на кшталт TLS 1.3 у мостах CAN-Ethernet забезпечує надійну аутентифікацію та захищений канал передачі між різнорівневими сегментами мережі, зберігаючи при цьому малу затримку за рахунок оптимізованих набутків шифрування.

Крім сертифікації окремих компонентів, важливою практикою є проходження повного аудиту кібербезпеки системи сторонніми фахівцями. Такий аудит включає перевірку реалізації криптографічних протоколів, оцінювання захищеності ключових матеріалів, аналіз можливих векторів атак та проведення пентестингу в умовах, максимально наближених до експлуатаційних. Позитивні результати такого аудиту стають вагомим аргументом для виробників автомобілів і Tier-1 постачальників у контексті доведення відповідності продукції стандартам безпеки.

3.5.3. Інтеграція та сумісність (AUTOSAR, CAN FD)

Криптографічні бібліотеки, такі як ESCRYPT CycurLIB і AUTOSAR J1939 Security Extensions, мають повну інтеграцію з Classic Platform, зокрема через модулі Crypto Service Manager, Crypto Driver та PDU Router. Вони дозволяють реалізовувати механізми автентифікації повідомлень (MAC), перевірки актуальності (freshness counters), а також керування ключами у відповідності до моделей Key Management Service (KMS). Це забезпечує не лише функціональну сумісність, а й відповідність до вимог ISO 26262 і ISO/SAE 21434.

Система виявлення аномалій ESCRYPT CycurIDS також передбачає інтеграцію в AUTOSAR-середовище через інтерфейси до діагностичних та логічних модулів (DEM, DCM), що дозволяє здійснювати моніторинг трафіку у реальному часі без порушення архітектурної цілісності ECU. Аналогічним чином, AUTOSAR SecOC, як стандартний компонент специфікації, забезпечує повну сумісність із транспортними модулями CAN, CAN FD, FlexRay та Ethernet.

Окремої уваги заслуговує питання сумісності апаратних CAN-фільтрів із середовищем AUTOSAR. Наприклад, шлюзи IXXAT CAN@net NT 420 та NXP S32K3 Gateway можуть бути інтегровані як периферійні пристрої, що забезпечують фізичне відділення мереж із різними рівнями довіри. За рахунок підтримки внутрішньої маршрутизації та фільтрації кадрів на основі конфігурацій AUTOSAR CAN Network Configuration, ці пристрої можуть працювати як транзитні вузли без порушення логіки обміну повідомленнями між ECU.

Сумісність засобів захисту з CAN FD є важливою характеристикою при оцінці їхньої придатності до сучасних вимог. Зокрема, CANcrypt повністю підтримує роботу з CAN FD, реалізуючи розширені механізми автентифікації та шифрування з мінімальним впливом на продуктивність. Додатково, оптимізація на рівні фреймворків дозволяє використовувати змінні швидкості передачі залежно від типу кадру, зберігаючи високу ефективність передачі захищених повідомлень.

NXP S32K3 Security Gateway Module, орієнтований на роботу з CAN FD, має вбудовані криптографічні прискорювачі (HSE – Hardware Security Engine), що забезпечують апаратну підтримку алгоритмів AES, SHA-2 та ECC. Це дозволяє обробляти зашифровані повідомлення на високій швидкості без перевантаження основного CPU. Також шлюз підтримує мультиканальну конфігурацію, зокрема одночасну роботу з декількома CAN FD-мережами та LIN-інтерфейсами, що є типовим для автомобілів з великою кількістю підсистем.

Системи IDS/IPS, такі як Vector CANoe, забезпечують повноцінну підтримку CAN FD як у режимі емуляції, так і під час інтеграції з реальними транспортними стек-бібліотеками. Це дозволяє проводити тестування з урахуванням усіх особливостей розширеного протоколу, включаючи налаштування фаз передачі, затримок синхронізації та валідацію довших кадрів. CANoe підтримує конфігурації з AUTOSAR через модулі RTE (Run-Time Environment) та COM Stack Simulation, що забезпечує безконфліктну роботу з CAN FD.

Також варто згадати AUTOSAR J1939 Security Extensions, які враховують специфіку CAN FD для реалізації Secure Communication між тягачем та причепами в системах комерційного транспорту. Цей модуль передбачає буферизацію кадрів, додавання верифікаційних даних та перевірку цілісності, враховуючи обмеження, накладені специфікацією J1939.

3.5.4. Вартість та ліцензування

Одним із апаратних рішень для захисту CAN-шини є Ixxat CAN@net NT 420. Пристрій підтримує чотири канали CAN, з яких два сумісні з CAN FD, та забезпечує довготривалу комунікацію з функціональністю обробки повідомлень. Ціна пристрою становить приблизно €899–€1,089 (≈ $978–$1,185), залежно від постачальника та умов доставки. Придбання апаратного пристрою включає необхідне програмне забезпечення для налаштування та використання, додаткові ліцензії не вимагаються.

Іншим апаратним рішенням є мікроконтролери сімейства S32K3 від NXP, які пропонують 32-бітові рішення на базі Arm Cortex-M7 з різними конфігураціями ядер, пам'яті та периферійних пристроїв. Вони забезпечують високу продуктивність та функціональну безпеку відповідно до стандарту ISO 26262 до рівня ASIL D. Референсний дизайн S32K3-T-BOX доступний за ціною від $550 до $579.11, залежно від постачальника. Придбання апаратного модуля включає доступ до безкоштовного програмного забезпечення S32 Design Studio, яке підтримує як AUTOSAR, так і non-AUTOSAR драйвери. Додаткові ліцензії можуть знадобитися для використання певних функцій або модулів.

Серед систем виявлення та запобігання вторгнень слід відзначити Vector CANoe – потужний інструмент для аналізу та емуляції мережевих протоколів, включаючи CAN, LIN та FlexRay. Він широко використовується для розробки, тестування та діагностики вбудованих систем. Базова ліцензія CANoe PRO коштує приблизно $2,450. Додаткові модулі, такі як підтримка UDS або CAPL, можуть додати $500-$1,000 до вартості. Повна конфігурація з усіма модулями може досягати $8,000-$10,000. Ліцензії надаються на основі модульного підходу, де кожен додатковий модуль вимагає окремої ліцензії. Ліцензії можуть бути прив'язані до конкретного апаратного ключа або комп'ютера.

Ще одним рішенням є ESCRYPT CycurIDS є вбудована система виявлення вторгнень, яка ідентифікує, повідомляє та реєструє атаки та аномалії в автомобільних мережах, таких як CAN та Ethernet. Рішення може бути адаптоване до конкретного транспортного засобу з використанням розподілених та хост-базованих IDS-сенсорів, менеджера та репортера. Ціни на ліцензії надаються за запитом – вони залежать від кількості захищених вузлів та обсягу функціональності. Ліцензії надаються на основі кількості захищених вузлів та можуть включати різні компоненти системи. Умови ліцензування можуть варіюватися залежно від конкретних потреб замовника.

У сфері криптографічних бібліотек CANcrypt є рішенням для забезпечення безпеки в мережах CAN, яке поєднує масштабовані функції безпеки для CAN. Програмне забезпечення розроблене для прототипування та початкового

пілотного виробництва. Базова ліцензія коштує €175 (≈ $190) і включає комерційну версію програмного забезпечення. Ліцензія дозволяє використання програмного забезпечення для прототипування та початкового пілотного виробництва до 500 пристроїв. Для масового виробництва необхідно отримати додаткову ліцензію.

ESCRYPT CycurLIB – це комплексна та налаштовувана криптографічна бібліотека, яка забезпечує високу продуктивність у середовищах з обмеженими ресурсами. Вона підтримує всі поширені криптографічні алгоритми та стандарти сертифікатів. Ціни на ліцензії надаються за запитом – вони залежать від кількості захищених вузлів та обсягу функціональності. Ліцензії надаються на основі кількості захищених вузлів та можуть включати різні компоненти бібліотеки. Умови ліцензування можуть варіюватися залежно від конкретних потреб замовника.

Серед open-source інструментів CANtact – це апаратний інтерфейс для підключення до мережі CAN через USB, а can-utils – набір утиліт для роботи з CAN в Linux-середовищі. Апаратна частина (CANtact) коштує приблизно $60–$100, програмне забезпечення (can-utils) є безкоштовним. Програмне забезпечення з відкритим вихідним кодом, доступне під ліцензією GNU General Public License (GPL).

Також can-isotp та cantools – це бібліотеки для роботи з протоколами CAN та ISO-TP у середовищі Python. Це безкоштовні бібліотеки з відкритим вихідним кодом. Вони доступні під ліцензіями MIT або BSD, що дозволяє вільне використання, модифікацію та розповсюдження.

Порівнюючи вартість та ліцензування різних засобів захисту CAN-шини, можна зробити висновок, що апаратні рішення, такі як Ixxat CAN@net NT 420 та NXP S32K3-T-BOX, мають фіксовану вартість придбання та не вимагають додаткових ліцензій для базового використання. Системи виявлення та запобігання вторгнень, такі як Vector CANoe та ESCRYPT CycurIDS, мають вищу вартість та складніші умови ліцензування, що залежать від обсягу функціональності та кількості захищених вузлів. Криптографічні бібліотеки, як-

от CANcrypt та ESCRYPT CycurLIB, пропонують гнучкі умови ліцензування, залежно від етапу розробки та обсягу використання. Open-source інструменти є безкоштовними та доступними для широкого кола користувачів, проте можуть вимагати додаткових зусиль для інтеграції та підтримки.

ВИСНОВКИ

У ході дослідження теми захисту CAN-шини було встановлено, що, незважаючи на її широке використання в автомобільній промисловості, вона має низку вразливостей, зумовлених відсутністю вбудованих механізмів безпеки, таких як автентифікація та шифрування. Це створює потенційні загрози, зокрема можливість перехоплення та модифікації повідомлень, атаки типу «відмова в обслуговуванні» (DoS), а також експлуатацію бездротових інтерфейсів для несанкціонованого доступу до мережі транспортного засобу.

Аналіз існуючих методів захисту CAN-шини показав, що найбільш ефективними є комплексні підходи, які поєднують фізичне сегментування мережі, використання міжмережевих екранів (CAN-фаєрволів), впровадження систем виявлення та запобігання вторгнень (IDS/IPS), а також застосування криптографічних засобів для забезпечення цілісності та автентичності переданих даних. Крім того, важливу роль відіграє логування подій та резервування каналів зв'язку для забезпечення безперервності роботи критичних систем.

Огляд комерційних та open-source засобів захисту показав, що вибір конкретного рішення залежить від вимог до продуктивності, ступеня безпеки, сумісності з існуючими стандартами (наприклад, AUTOSAR, CAN FD), а також від вартості та умов ліцензування. Комерційні рішення, як-от IXXAT CAN\@net NT 420 або NXP S32K3 Security Gateway Module, забезпечують високий рівень інтеграції та підтримку сучасних стандартів безпеки, але можуть бути дорогими. Натомість open-source інструменти, такі як CANtact з can-utils або бібліотеки can-isotp та cantools для Python, є доступними та гнучкими, але вимагають додаткових зусиль для інтеграції та підтримки.

Узагальнюючи, для забезпечення надійного захисту CAN-шини необхідно впроваджувати багаторівневі стратегії, що поєднують апаратні та програмні засоби, з урахуванням специфіки конкретного застосування та потенційних загроз.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. CSS Electronics. CAN Bus Explained – A Simple Intro [Електронний ресурс]. URL: https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial.
2. European Space Agency. CAN – Controller Area Network Bus [Електронний ресурс]. URL: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Onboard_Computers_and_Data_Handling/CAN_-_Controller_Area_Network_Bus.
3. European Cooperation for Space Standardization. ECSS-E-ST-50-15C – CANbus extension protocol. 01.05.2015 [Електронний ресурс]. URL: https://ecss.nl/standard/ecss-e-st-50-15c-space-engineering-canbus-extension-protocol-1-may-2015/
4. International Organization for Standardization. ISO 11898-1:2024 – Road vehicles — Controller area network (CAN). 2024 [Електронний ресурс]. URL: https://www.iso.org/standard/86384.html
5. CAN in Automation. [Електронний ресурс]. URL: https://www.can-cia.org/
6. Deng R., Wang Y., Chen D., Wei J., Ding S. An Automotive CAN Bus Safety System Based On Safety Level / IEEE Transactions on Vehicular Technology, 2023 [Електронний ресурс]. URL: https://ieeexplore.ieee.org/document/10242774
7. CAN in Automation. CAN in Automation – Mercedes W140: First car with CAN [Електронний ресурс]. URL: https://can-newsletter.org/engineering/applications/160322_25th-anniversary-mercedes-w140-first-car-with-can
8. Albert A. Comparison of Event Triggered and Time Triggered Concepts with Regard to Distributed Control Systems / Robert Bosch GmbH. Embedded World, Нюрнберг, 2004 [Електронний ресурс]. URL: https://archive.air.in.tum.de/pub/Main/TeachingWs2013MSE/embedded_world_04_albert.pdf

9. NISMO. NISMO Increases GT6 GPS Data Logger Functionality and Track Count [Електронний ресурс]. URL: https://www.gtplanet.net/tag/gps-data-logger
10. Nasser A. M. K. Automotive Cybersecurity Engineering Handbook: the automotive engineer's roadmap to cyber-resilient vehicles / Ahmad M. K. Nasser. Birmingham–Mumbai : Packt Publishing Ltd., 2023. – 432 c. ISBN 978-1-80107-653-1
11. Palmer Z. Thieves are now stealing cars via a headlight "CAN injection" / Autoblog, 18.04.2023 [Електронний ресурс]. URL: https://www.autoblog.com/carbuying/vehicle-headlight-can-bus-injection-theft-method-update
12. Daigmorte H., Boyer M. Evaluation of admissible CAN bus load with weak synchronization mechanism / Proc. 24th Int. Conf. on Real Time Networks and Systems (RTNS 2017), Гренобль, 2017 [Електронний ресурс]. URL: https://openscience.isae-supaero.fr/digitalCollection/DigitalCollectionAttachmentDownloadHandler.ashx?documentId=12503
13. IOActive. Transportation & Vehicle [Електронний ресурс]. URL: https://ioactive.com/industry/transportation-vehicle/
14. Voss W. Controller Area Network (CAN Bus) – Message Frame Architecture / Copperhill Technologies, 21.11.2018 [Електронний ресурс]. URL: https://copperhilltech.com/blog/controller-area-network-can-bus-message-frame-architecture/
15. Jindhira Pooja S. CAN Standard Data Frame Format / Medium, 25.06.2024 [Електронний ресурс]. URL: https://medium.com/@sjindhirapooja/can-standard-data-frame-format-846b8f9fc749
16. National Instruments. CAN Frames [Електронний ресурс]. URL: https://www.ni.com/docs/en-US/bundle/ni-xnet/page/can-frames.html
17. University of Michigan. EECS 461: CAN Notes [Електронний ресурс]. URL: https://www.eecs.umich.edu/courses/eecs461/doc/CAN_notes.pdf

18. Software Engineering Institute. CAN Bus Security: A Survey. 2016 [Електронний ресурс]. URL: https://insights.sei.cmu.edu/documents/472/2016_019_001_453877.pdf
19. EEVblog. A Naive Question About Automotive CANbus Sniffing and Spoofing [Forum post] [Електронний ресурс]. URL: https://www.eevblog.com/forum/projects/a-naive-question-about-automotive-canbus-sniffing-and-spoofing/
20. Purdue University. CAN Bus Protocol Overview [Електронний ресурс]. URL: https://www.cs.purdue.edu/homes/bb/mubark1.pdf
21. Vector Informatik GmbH. Reliably Detecting and Defending Against Attacks: Requirements for Automotive Intrusion Detection Systems [Електронний ресурс]. Vector Informatik GmbH. Режим доступу: https://cdn.vector.com/cms/content/know-how/_technical-articles/Security_Intrusion_Detection_AutomobilElektronik_202003_PressArticle_EN.pdf
22. National Center for Biotechnology Information. PMC10575265 [Електронний ресурс]. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC10575265/
23. KICS Journal. Understanding CAN Bus Vulnerabilities and How Blockchain Can Amplify Security. 2020 [Електронний ресурс]. URL: https://journal-home.s3.ap-northeast-2.amazonaws.com/site/2020kics/presentation/0128.pdf
24. Southwest Research Institute. Zero Trust Architecture for Automotive Networks, 10 R6352 [Електронний ресурс]. URL: https://www.swri.org/what-we-do/internal-research-development/2024/automotive-transportation/zero-trust-architecture-automotive-networks-10-r6352
25. International Organization for Standardization. ISO 11898 1:2015 – Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling [Електронний ресурс]. URL: https://www.iso.org/standard/86384.html

26. Nature Communications. A Comprehensive Study on CAN Bus Security. 2025 [Електронний ресурс]. URL: https://www.nature.com/articles/s41598-025-98433-x
27. NDSS Symposium. VehicleSec 2024: CAN Bus Security Challenges. 2024 [Електронний ресурс]. URL: https://www.ndss-symposium.org/wp-content/uploads/vehiclesec2024-43-paper.pdf
28. Zhang Y. CAN Bus Security: A Survey. 2025 [Електронний ресурс]. URL: https://www.tu-chemnitz.de/informatik/CAS/publications/pdfs/Zhang2025a.pdf
29. Wen H. Securing CAN Bus Communications. USENIX Security Symposium, 2020 [Електронний ресурс]. URL: https://www.usenix.org/system/files/sec20summer_wen_prepub.pdf
30. Jindhira Pooja S. Understanding CAN Bus Vulnerabilities and How Blockchain Can Amplify Security / Medium, 25.06.2024 [Електронний ресурс]. URL: https://medium.com/@chaincom/understanding-can-bus-vulnerabilities-and-how-blockchain-can-amplify-security-a58388bf1fb4
31. Zhang Y. CAN Bus Security: A Survey / arXiv, 2020 [Електронний ресурс]. URL: https://arxiv.org/pdf/2006.08723
32. NCC Group. Latest Threats to the Connected Car and Intelligent Transport Ecosystem [Електронний ресурс]. URL: https://www.nccgroup.com/media/ns3fldjd/_latest-threats-to-the-connected-car-and-intelligent-transport-ecosystem.pdf
33. Staro V. VTC 2016: CAN Bus Security / Boston University, 2016 [Електронний ресурс]. URL: https://people.bu.edu/staro/VTC_2016.pdf