Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Безпеки інформаційних технологій
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Використання децентралізованих систем для захисту автомобільних систем
(тема)

Виконав: Фесенко Д. О.
(прізвище, ініціали)
студент 2 курсу, групи БІКСм-19-1
Спеціальність 125 Кібербезпека
(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма «Безпека інформаційних і комунікаційних систем»
(повна назва освітньої програми)

Керівник проф. Горбенко І.Д.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Халімов Г.З.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Безпеки інформаційних технологій
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 125 Кібербезпека
(код і повна назва)

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

Освітня програма «Безпека інформаційних і комунікаційних систем»
(повна назва)

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

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові *Фесенка Дмитра Олександровича*
(прізвище, ім'я, по батькові)

1. Тема роботи *Використання децентралізованих систем для захисту автомобільних систем*
затверджена наказом по університету від "22" жовтня2020 р. № 1412Ст
2. Термін подання студентом роботи (проекту) 15.12.2020
3. Вихідні дані до роботи (проекту) статі щодо використання технології блокчейн для систем авдентифікації
4. Зміст пояснювальної записки (перелік питань, що потрібно розробити)
1. Розглянути безпечність сучасних систем авдентифікації автомобільних сис-тем
2. Розглянути вимоги та модель безпеки для надійного та безпечного каналу передачі даних
3. Створити архітектуру захищеної системи авдентифікації для автомобільних систем з використанням системи блокчейн
4. Розробити модель системи безпеки автомобіля
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій *Презентаційний матеріал у вигляді слайдів*
6. Основна література та джерела. Lemke K. Embedded Security in Cars: Securing Current and Future // сб. − 2017 – С. 430.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської атестаційної роботи	Термін виконання етапів роботи	Примітка
1	*Отримання завдання*	*05.09.20*	
2	*Аналіз літературних джерел за темою атестаційної роботи*	*05.09.20-28.09.20*	
3	*Розгляд систем автентифікації авто-мобіля*	*28.09.20-15.10.20*	
4	*Використання блокчейну для систем автентифікації автомобіля*	*15.10.20-11.11.20*	
5	*Програмна реалізація системи авдентифікації авто-мобіля*	*11.11. 20-25.11. 20*	
6	*Оформлення пояснювальної записки*	*25.11. 20-04.12. 20*	
7	*Представлення роботи на здачу*	*04.12.20-16.12.20*	

Дата видачі завдання _____ _____20__ р.

Студент _____
(підпис)

Керівник роботи (проекту) _____ __проф. Горбенко І.Д._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до роботи містить 100 сторінок, 44 рисунки, 3 таблиці, 11 посилань.

Метою атестаційної роботи є розробка системи безпечної автентифікації під час отримання правомірного доступу до автомобіля.

Об’єкт дослідження – процеси автентифікації для доступу до автомобіля та його систем.

Предмет дослідження – атаки на систему безпеки контролю доступу до автомобіля та методи захисту від них з використанням децентралізованих технологій.

Відповідно до поставленої мети у роботі вирішуються такі задачі:

1) Розгляд безпечності сучасних систем автентифікації, що використовуються для автомобільних систем;

2) Обгрунтування, вибір та аналіз побудування захищеної системи автентифікації автомобіля на основі розглянутих систем безпеки;

3) Вимоги та модель безпеки для каналу передачі даних для системи безпеки автомобіля для унеможливлення атак на перехоплення, підміну, модифікацію даних;

4) Створення архітектури захищеної системи автентифікації для автомобільних систем з використанням мережі блокчейн;

5) Аналіз та порівняння захищеності розробленої архітектури системи захисту з комерційними аналогами;

6) Розроблена програмна модель роботи системи безпеки;

7) Розроблена програмна модель комплексу для проведення атак на системи автентифікації автомобіля для перехоплення та підміни даних;

Ключові слова: ПОРУШНИК, АВТОМОБІЛЬ, IDS, АТАКА, МЕРЕЖА, БЛОК, АВТЕНТИФІКАЦІЯ, СИГНАЛІЗАЦІЯ, ПОПЕРЕДЖЕННЯ ВТОРГНЕНЬ, БЛОКЧЕЙН, СТІЙКІСТЬ, МЕРЕЖА.

ABSTRACT

The explanatory note contains: 100 pages, 44 figures, 3 tables, 11 sources.

The purpose of the work is to develop a system of secure identification when obtaining legal access to the car, the development of proposals for cryptographic protection of information in the data network for car identification systems.

The object of this research is the authentication processes for access to the car and its systems.

The subject of the research is attacks on the security system of access control to the car and methods of protection against them with the use of decentralized technologies.

The paper considers the structure of building access control systems for vehicles, network security monitoring system, which identifies users to access the vehicle and feedback to update data on the status of the car security system.

In accordance with the set goal, the following tasks are solved in the work:

1) Consideration of security of modern identification systems used for automotive systems;

2) Substantiation, selection and analysis of the construction of a secure vehicle identification system based on the considered security systems;

3) Requirements and security model;

4) Creating a secure authentication system architecture for automotive systems using a blockchain network;

5) Analysis and comparison of security of the developed architecture of the protection system with commercial analogues;

6) Developed a software model of the security system;

7) Developed a software model of the complex for attacks on vehicle identification systems for interception and data corruption;

Key words: VIOLATOR, CAR, IDS, ATTACK, NETWORK, BLOCK, AUDENTIFICATION, SIGNALING, INVASION PREVENTION.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА СИМВОЛІВ

НСД – Несанкціонований доступ
АС – Автоматизована система
БЛК – Блок
ІС – Інформаційна система
MITM – man-in-the-middle (атака типу «людина посередині»).
ID – identifier (ідентифікатор).
SHA – Secure Hash Algorithm
CAN – Controller area network (мережа контролерів)
ІТС – Інформаційно-телекомунікаційна система
КЗЗ – Комплекс засобів захисту
КЗІ – Криптографічний захист інформації
КСЗІ – Комплексна система захисту інформації
КТЗ – Комплекс технічних засобів
ЛОМ – Локальна обчислювальна мережа
НД – Нормативний документ
ОІД – Об’єкт інформаційної діяльності
ОС – Операційна система
ПЕОМ – персональна електронно-обчислювальна машина
ПЗ – Програмний засіб
ПК – Програмний комплекс
РС – Робоча станція
СТО – Станція технічного огляду
ТІ – Технічна інформація

ВСТУП

Успішно проведені атаки на автомобільні комп’ютерні системи стають реальністю, збільшується кількість загроз та атак спрямованих на транспортні засоби наступного покоління та їх вузли та механізми. Транспортні засоби, що обладнані засобами доступу до мережі інтернет відкривають великі можливості для створення нових технологічних та зручних засобів, що спрощують життя людини, але водночас це відкриває нові можливості для зловмисників та загроз інформаційної безпеки[2].

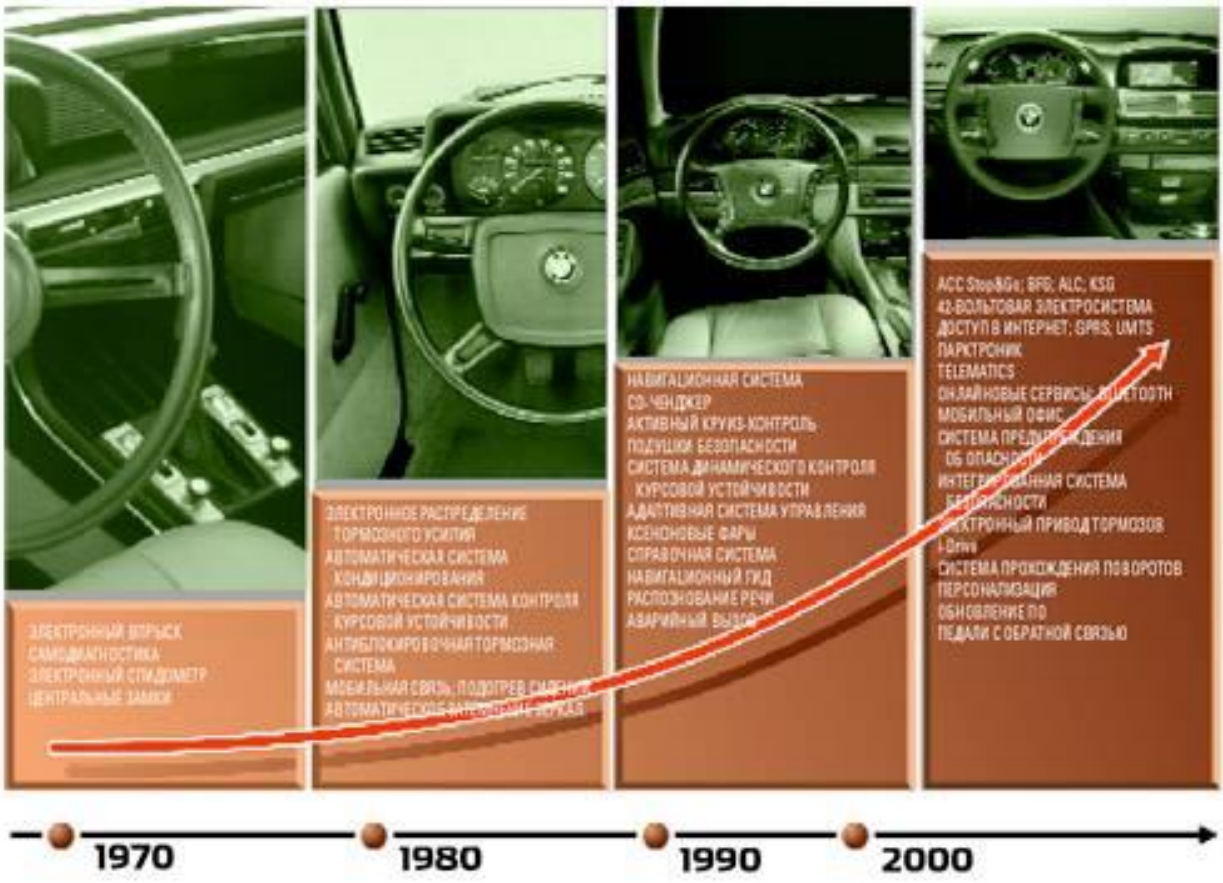

Рисунок 1. – Тенденції розвитку інтеграції комп’ютерних систем в автомобілі

З огляду на тенденції розвитку, що зображені на рисунку 1, а до кінця двадцятих років другого тисячоліття усі нові автомобілі будуть мати в собі безліч електронних систем, при цьому безпека використання вимагає детального розгляду захищеності, оскільки злам та модифікація електронних систем автомобіля

може дозволити отримати неправомірний доступ керування транспортним засобом, що дуже цікавить зловмисників і є досить реальною і постійно зростаючою проблемою для безпеки автомобільних систем[5]. Різні компанії намагаються створити адаптивний круїз-контроль, який відповідає за автоматичне керування автомобіля. Він повинен стабілізовувати швидкість, а також використовує радар для виявлення предметів на дорозі для визначення, що в небезпосередній близькості знаходиться інший автомобіль, тоді система підтримує безпечну відстань. Протягом наступного десятиліття слід очікувати появи систем більш високого рівня автоматизації водіння, наприклад, програма HAVEit, спрямована на створення високоавтоматизованих автомобілів, частково фінансується Європейською комісією. В рамках цього проекту був розроблений і випробуваний автопілот, який дозволяє автомобілю рухатися зі швидкістю до 130 км / год, контролюючи поперечний і поздовжнє переміщення за допомогою асистента збереження смуги руху і адаптивного круїз-контролю.

Рисунок 2. – Концепт нового автомобіля з автономним автопілотом від Google

Автопілот розпізнає швидкісні обмеження, виявляє ділянки ведення дорожніх робіт і підтримує необхідну відстань від попереду автомобілів, що йдуть, утримуючи при цьому смугу руху і зберігаючи задану водієм швидкість. За повідомленням концерну Volkswagen, серійне виробництво подібних систем може початися через п'ять років.

В Google розробили концепт автомобіля з автоматичним рульовим керуванням, оснащені сенсорами, навігаційними приладами і штучним інтелектом(зовнішній вигляд зображений на рисунку 2). Ці машини розраховані на багатокілометровий пробіг, пройшли польові випробування, під контролем автоінженерів. З боку користувачів використання допоміжних електронних систем – це більш зручні та високотехнологічні засоби, якими користуватися одне задоволення, оскільки такі системи дозволяють користувачу моніторити стан вузлів авто в режимі реального часу, дивитись фільми або працювати з іншими даними та навіть передавати їх по P2P мережі автомобіля. Це все значні плюси для бізнесу, також для виробників частин вузлів автомобілів, частин цих вузлів та інших взаємозв’язаних з ними видів бізнесу, що в сукупності можуть дати водію саме те, що йому потрібно. Таким чином такі автомобілі стають важливою частиною життя покупця, що робить інформаційну безпеку важливим аспектом безпеки життя покупця[2]. Сьогодні, коли децентралізовані технології розвивається дуже стрімко, і на їх основі з'являються нові технології, що призвані замінити багато сучасних речей до яких світ встиг звикнути, що дозволяють відійти від централізованої обробки даних: від систем зберігання користувацьких даних до державних систем голосування, проте далеко не всі розробники у повній мірі розуміють її можливості та недоліки в плані безпеки, адже ця тема мало вивчена, оскільки нові технології з'являються настільки часто, що фахівці не встигають зреагувати на їх появу та провести аналіз їх захищеності як необхідно[1].

Блокчейн – це постійні системи цифрових регістрів, реалізовані розподіленим чином (тобто, без центрального сховища) та не мають центрального органу[3].

Система блокчейн цікава тим, що на протязі десятиліття створює резонанс навколо людей як система яка є децентралізованою – тобто така, яка не має вузол, який реалізує критично важливі для системи операції. Це в першу чергу робить систему типу блокчейн теоретично стійкою до будь-якої атаки типу DDoS. По-друге система блокчейн використовує криптографічні засоби захисту інформації. На сьогодні існують безліч систем грошового обігу на базі системи блокчейн, або так звані криптовалютні системи. На основі технології блокчейн будуються «розумні» контракти, реєстри, нотаріати і т. і. Гіпотетично, система блокчейн у глобальному сенсі позиціонується як система яка має вирішити проблеми пов'язані з централізованими системами – це проблеми корупційності, проблеми підробки даних у системах, проблеми доступу до даних, проблеми атак на централізоване сховище даних. Разом з тим система блокчейн дозволяє побудувати відкриту для усього людства систему документу-обігу.

Через можливості неспростовності та захищеності дану технологію можна використовувати для систем безпеки автомобіля, використовуючи захищеність системи передачі та зберігання даних для створення засобів та систем автентифікації, оскільки проекти, що розробляють системи з застосуванням технології блокчейн, отримують стан можливість впевнено завіряти, що данні користувача будуть отримуватися саме власником, а не іншими особами та не можуть бути змінені під час роботи[6]. Але, щоб досягти гарного рівня захищеності з використанням блокчейну необхідно буде розробити гарну програмну реалізацію серйозно розробленої архітектури з виконанням всіх заходів безпеки для розробки таких програмних засобів.

1 ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕНЬ

У рамках виконання даної роботи повинні бути розглянуті наступні питання:

1) Розгляд безпечності сучасних систем автентифікації, що використовуються для автомобільних систем;

2) Обгрунтування, вибір та аналіз побудування захищеної системи автентифікації автомобіля на основі розглянутих систем безпеки;

3) Визначення вимог та опис моделі безпеки для каналу передачі даних для системи безпеки автомобіля для унеможливлення атак на перехоплення, підміну, модифікацію даних;

4) Створення архітектури захищеної системи автентифікації для автомобільних систем з використанням мережі блокчейн;

5) Аналіз та порівняння захищеності розробленої архітектури системи захисту з комерційними аналогами;

6) Розробка програмної моделі роботи системи безпеки;

7) Розробка програмної моделі комплексу для проведення атак на системи автентифікації автомобіля для перехоплення та підміни даних;

Повинні бути представлені:

1) Вимоги до рівня безпеки, довіри та захищеності для системи автентифікації;

2) Принципи побудови захищених протоколів автентифікації на основі децентралізованих технологій;

3) Пропозиції щодо забезпечення відповідного рівня безпеки;

4) Аналіз розробленої архітектури та її програмної реалізації.

Головне завдання роботи – є розроблення неспростовної системи автентифікації з урахуванням відповідних загроз та вимог з результатом у вигляді програмної реалізації програмної моделі системи безпеки.

2 ХАРАКТЕРИСТИКА ОБ`ЄКТА ІНФОРМАЦІЙНОЇ ДІЯЛЬНОСТІ

2.1 Основні відомості про об’єкт інформаційної діяльності

Об’єктом інформаційної діяльності є абстрактний транспортний засіб, що використовує для захисту від несанкціонованого доступу систему безпеки високого класу з постійним моніторингом стану захищеності транспортного засобу за допомогою GSM зв’язку, використовуючи який дані відправляються на центральний сервер, до якого може отримати доступ власник для перегляду стану захищеності автомобіля у вигляді веб-сервісу, або з використанням програмного забезпечення, що встановлюється на смартфон[2]. Для прикладу такого транспортного засобу розглянемо автомобіль марки “Toyota” моделі ”Land Cruiser” 2020 року виготовлення.

Рисунок 2.1. – Схематичне зображення систем в об’єкті інформаційної діяльності

Сучасні автомобілі є складною мережею, що складаються з багатьох незалежних вбудованих обчислювальних систем, з'єднаних по мережі. Наприклад, автомобіль, що розглядається складається з понад 200

мікропроцесорів, на яких працює до 65 мільйонів рядків коду, що робить його одним з найскладніших програмних систем, на рисунку 2.1 показано більш детальне зображення складності систем ОІД. Як і у випадку з будь-якою складною системою програмного забезпечення, збереження безпеки є актуальною справою.

Вимоги безпеки, що висуваються до компонентів системи автентифікації – це вимоги цілісності, доступності та достовірності, оскільки для правильної роботи всіх системи безпеки автомобіля необхідно, щоб інформація, що курсує в ІС залишалася завжди доступною і цілісною та була не скомпрометованою, бо це є дуже важливим аспектом для правильного керування параметрами безпеки автомобіля та взагалі правильного функціонування всіх його систем.

В даній мережі циркулюють критичні дані у вигляді пакетів, що курсують від одного компонента системи до іншого. Таким чином, компоненти зв’язані між собою та можуть взаємодіяти, що дозволяє автомобілю рухатися та ефективно використовувати системи керування та безпеки.

2.2 Загальна структурна схема і склад ІС

Інформаційна система являє собою мікроконтролер з встановленими додатковими модулями зв’язку з мережею GSM та інтеграції пристрою до мережі CAN.

Рисунок. 2.2. – Архітектура та склад інформаційної системи

Загальна структурна схема зображена на рисунку 2.2, в якій зображені всі компоненти системи.

Система побудована на основі клієнт-серверної архітектури, що складається з мікроконтролера з доступом до мережі інтернет[3], сервера, що виконує роль системи зберігання даних, що курсують в системі, системи моніторингу даних для виявлення проблем безпеки і загроз, що у випадку позаштатної ситуації відправляє повідомлення на смартфон власника транспортного засобу.

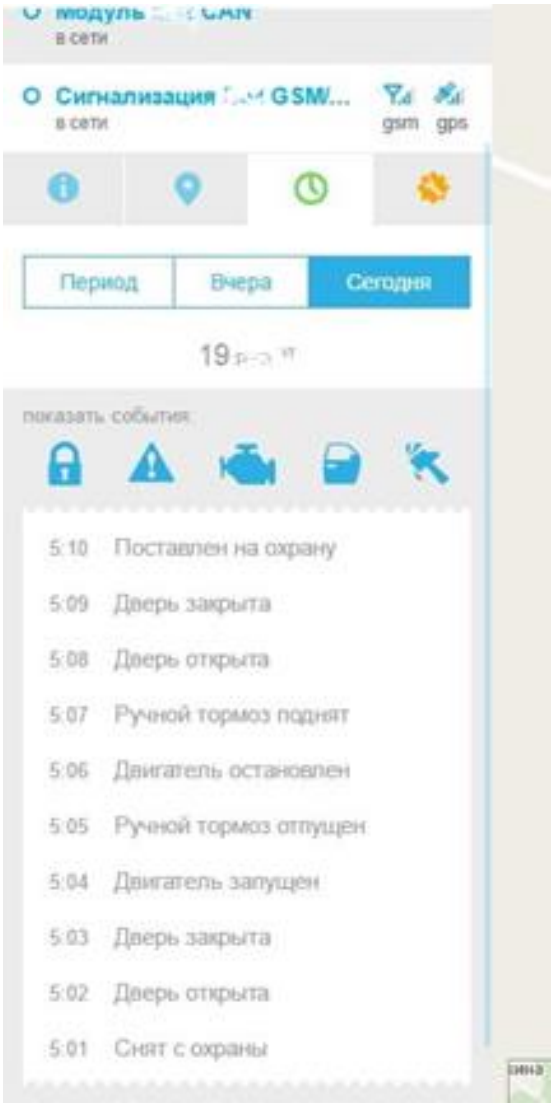

Рисунок 2.3 – Веб-інтерфейс системи моніторингу стану захищеності автомобіля

На рисунку 2.3 зображено інтерфейс системи моніторингу, використовуючи який власник транспортного засобу може відстежити стан захищеності його автомобіля використовуючи веб браузер.

2.3 Криптографічний захист системи передачі даних IC

В системі безпеки, що встановлений на об’єкт інформаційної діяльності в якості криптографічного захисту інформації, використовується KeeLoq, блоковий шифр, заснований на програмному компоненті "NLFSR", що являє собою регістр зсуву з нелінійним зворотним зв'язком. Односпрямований протокол передачі команди був розроблений Фредеріком Брувером, який є генеральним директором компанії Nanoteq Pty Ltd.

Алгоритм KeeLoq був розроблений в середині 80-х Джідеоном Куном разом з Вілліема Смитта в Nanoteq Pty Ltd (підрозділ Південної Африки) і був проданий Microchip Technology, Inc. в 1995 році за 10 млн. доларів. Алгоритм являє собою «плаваючий код», кодується і декодується за допомогою мікросхем NTQ105 / 106/115 / 125D / 129D і HCS2XX / 3XX / 4XX / 5XX.

Шифрування відбувається блоками по 32 біта з використанням 64 бітного ключа, один блок тексту шифрується за 528 раундів. Функція NLF є нелінійним зворотним зв'язком, яка приймає значення 0x3A5C742E або

$$F(a,b,c,d,e) = d \oplus e \oplus ac \oplus ae \oplus bc \oplus be \oplus cd \oplus de \oplus ade \oplus ace \oplus abd \oplus abc \quad (1)$$

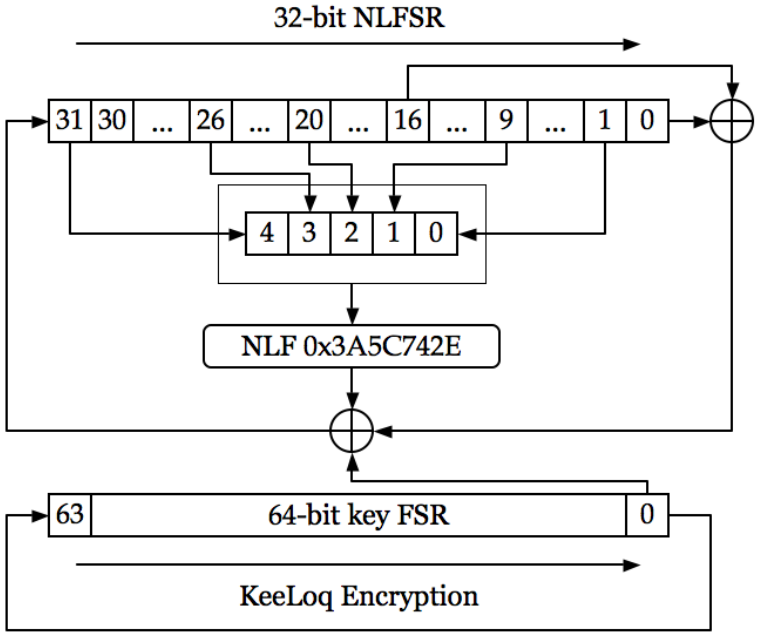

Рисунок 2.4 – Шифрування

NLF 0x3A5C742E – feedback function, F

F(a,b,c,d,e) = d ⊕ e ⊕ ac⊕ ae ⊕ bc ⊕ be ⊕ cd ⊕ de ⊕ ade ⊕ ace ⊕abd⊕abc(2)

Feedback:

$\varphi = F\,(yi31,yi26,yi20,yi9,yi1) \oplus yi16 \oplus yi0 \oplus ki0;$ (3)

Text: $Ri + 1 = (\varphi, yi31, \dots, yi1);$; (4)

Key: $Ki + 1 = (ki0, ki63, \dots, ki1);$; (5)

Алгоритм використовує 1, 9, 20, 26 і 31 біти з NLFSR для виведення під час шифрування і 0, 8, 19, 25 і 30 біти під час розшифрування. На виході виконується операція XOR з двома з бітів стану NLFSR (біти 0 і 16 на шифруванні і 31 і 15 біти на розшифровці) і з ключовим бітом (біт 0 з ключового стану на шифруванні і біт 15 з ключового стану на розшифровці) і дана операція подається назад в стан NLFSR на кожному раунді[2].

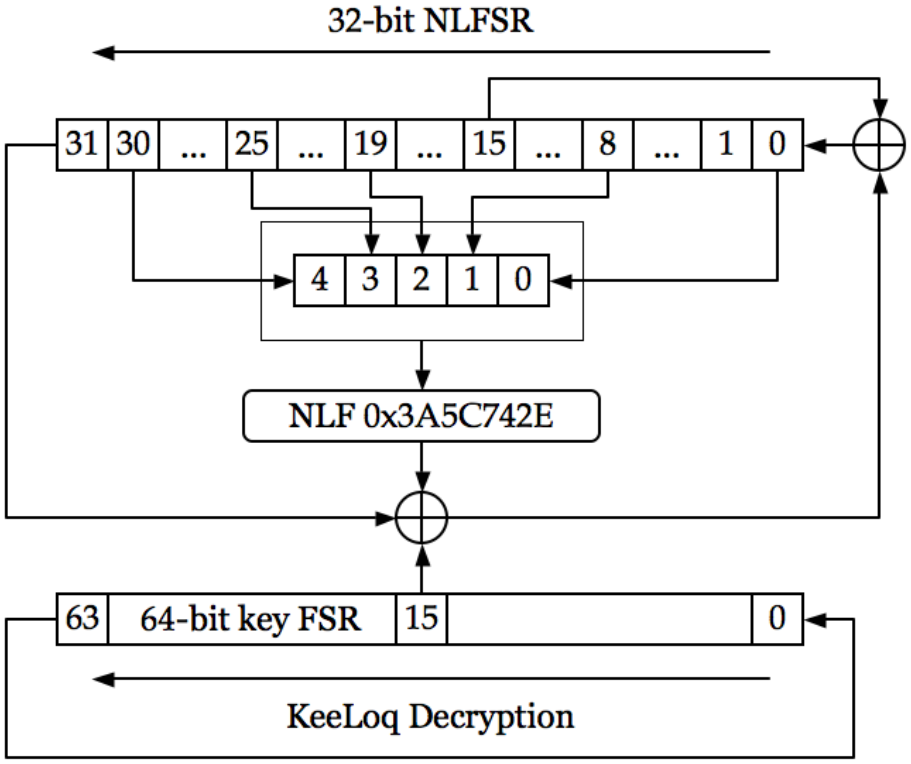

Рисунок 2.5 – Розшифрування

Feedback:

$$\varphi = F(yi30, yi25, yi19, yi8, yi0) \oplus yi15 \oplus yi31 \oplus ki15; \quad (1)$$

Text:

$$Ri + 1 = (yi30, \dots, yi0, \varphi); \quad (2)$$

Key:

$$Ki{+}1{=}(ki62,\dots,ki0,ki63); \quad (3)$$

KeeLoq вперше був проаналізований Андрієм Богдановим, який використовував метод «ковзної середньої» та ефективні лінійні наближення. Микола Куртуа атакував KeeLoq, використовуючи метод «ковзної середньої» та алгебраїчні методи. Атаки більш уразливі для «брутфорсу» ключового простору. Окрема реалізація «плаваючого коду» також часто уразлива для атаки з повторенням відправки пакетів, яка створює перешкоди на каналі, перериває і захоплюючи сам код і надалі збільшуючи час виконання в 4 рази від стандартного часу[3]. Ця вразливість KeeLoq дозволила створити так звані «грабери», популярні у викрадачів, які використовують мікросхеми FPGA для перебору основного ключа KeeLoq. Використовуючи деталі алгоритму, які витекли в широкі маси в 2006 році, дослідники почали вивчати вразливі місця алгоритму. Після визначення частини ключа, яка відповідає за певні моделі автомобіля, унікальний біт ключа може бути зламаний під час перехоплення синхронізації ключа і автомобіля[1].

2.4 Характеристики та можливості використання системи блокчейн для захисту ОІД

Блокчейн – це система регістрів, які являють собою розподілену систему та не мають центрального органу, що складаються з реєстрів обліку криптографічно підписаних транзакцій, згруповані в блоки, де кожен блок пов'язується з попереднім після перевірки[8].

Після додавання нових блоків попередні блоки стає все важче модифікувати, оскільки вони копіюються по всіх реєстрах обліку всередині мережі та будь-які конфлікти вирішуються автоматично за допомогою встановлених правил.

На високому рівні блокчейн використовують відомі механізми (пов'язані

списки, розподілені мережі) та криптографічні примітиви (гешування, цифрові підписи, публічні / приватні ключі).

Розглянемо можливість програмного моделювання роботи мережі блокчейн з використанням програмної реалізації.

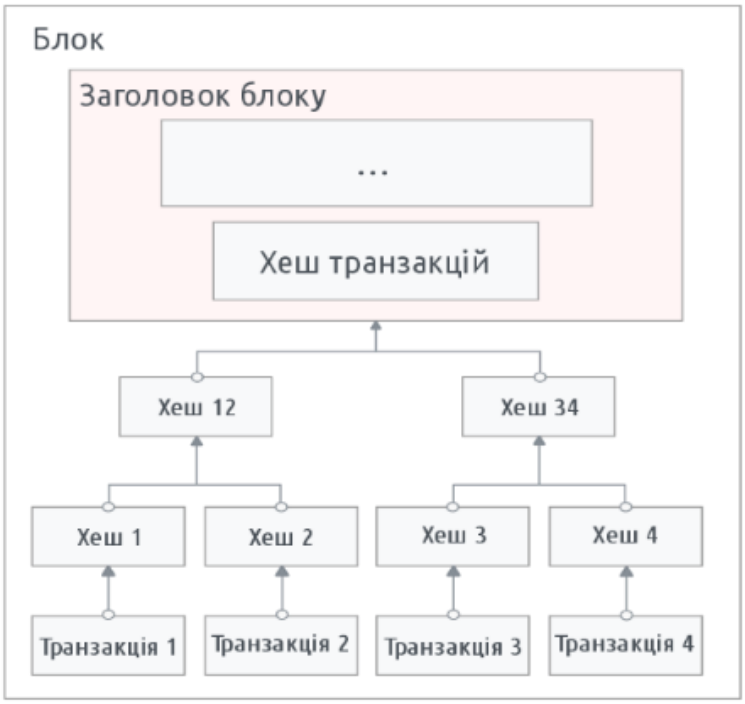

Рисунок 2.6. – Склад блоків в мережі блокчейн

Важливим елементом технології блокчейн є криптографічні геш-функції для багатьох операцій, таких як гешування вмісту блоку, склад яких показано на рисунку 2.6, що дозволяє обчислити відносно унікальних даних строку фіксованого розміру, яка дозволить перевірити цінності даних. Будь-яка зміна даних призведе до зовсім іншого вихідного геш – значення.

Алгоритм гешування, який використовується в багатьох технологіях блокчейн, – це SHA-2[10], який багато комп'ютерів підтримують цей алгоритм в апаратному забезпеченні, що робить його швидким для обчислення. Найчастіше в блокчейн-проектах використовуються реалізації з вихідним розміром 256 біт.

Таблиця 2.1 – Статистичні дані з імовірності колізії

Algorithm	The Size of the Message Digest (bit)	Message Block Size	Collision
MD2	128	128	Yes
MD4	128	512	Almost
MD5	128	512	Yes
RIPEMD	128	512	Yes
RIPEMD-128/256	128/256	512	No
RIPEMD-160/320	160/320	512	No
SHA-0	160	512	Yes
SHA-1	160	512	There is a Disability
SHA-256/224	256/224	512	No
SHA-512/384	512/384	1024	No
WHIRPOOL	512	512	No

Алгоритм гешування SHA-256 стійкий до колізій, оскільки для виявлення такої ситуації в SHA-256 необхідно виконати алгоритм в середньому близько 2^{128} разів. Технології блокчейн зберігають список транзакцій та створюють геш-значення для списку. Кожний учасник мережі, що має такий самий список транзакцій, може генерувати точно таке саме значення для перевірки. При зміні будь-якого значення транзакції в списку геш для цього блоку змінюється, що дозволяє легко виявити навіть незначні зміни в одному біті.

Транзакція – це запис передачі даних між сторонами-учасниками, що зберігається у вигляді, що представлений в таблиці 2.2 на прикладі передачі даних про нову транзакцію в мережі біткоїн. Кожен блок блокчейну містить декілька транзакцій.

Кожна транзакція вимагає для розгляду наступні інформаційні поля:

1) Дані – дані, що буди створені для передачі;
2) Ідентифікатор транзакції / геш-значення – унікальний ідентифікатор

для кожної транзакції. Деякі блокчейн системи використовують ідентифікатор, а інші приймають геш певної транзакції як унікальний ідентифікатор.

Рисунок 2.7 – Зріст користувачів мережі блокчейн для криптовалюти біткоїн

Важливо перевірити дійсність транзакції для уточнення вірності ствердження, що транзакція дійсно відбулася, тому всі транзакції підписані, та їх можна будь-коли перевірити за допомогою пари публічних / приватних ключів. Для організації зв'язку при передачі даних в мережах блокчейн використовуються адреси, які являють собою короткий буквено-цифровий рядок, що походить від відкритого (публічного) ключа користувача, використовуючи функцію гешування. Адреси використовуються для надсилання та отримання ифрових активів.

Адреси коротші за відкриті ключі та не є таємними. Для того, щоб створити адресу, це, як правило, означає взяття відкритого ключа, його гешування та перетворення гешу в текст.

Таблиця 2.2 – Приклад транзакції

	Input	Output	Amount	Total
Transaction ID: 0xa1b2c3	Account A	Account B	0.0321	
		Account C	2.5000	
				2.5321

Користувачі можуть генерувати необхідну кількість пар приватних / публічних ключів, таким чином і адрес, дозволяючи змінювати ступінь псевдоанонімності. Адреси виступають як ідентифікація блокчейну для користувача. Коли блокчейн розподіляє інформацію, це відбувається шляхом призначення їх на адресу. Підписавши цифрову транзакцію за допомогою приватного ключа, транзакцію можна перевірити за допомогою публічного ключа.

Централізовані реєстри обліку можуть мати недоліки, такі як:

1) Вони можуть бути втрачені або знищені: користувач повинен довіряти тому, що власник належним чином налаштував систему;

2) Транзакції можуть бути недійсними: користувач повинен довіряти тому, що власник ратифікує кожну отриману транзакцію;

3) Список транзакцій може бути не повним: користувач повинен довіряти, що власник включає всі чинні транзакції, які були отримані;

4) Дані транзакції можуть бути змінені: користувач повинен довіряти, що власник не змінює чинні операції.

Реєстр, що реалізований за допомогою блокчейну, може нівелювати ці проблеми завдяки використання розподіленого консенсусного механізму та того, що блокчейн реєстру обліку буде копіюватися та розповсюджуватись серед всіх вузлів всередині системи.

Користувачі можуть подавати кандидати транзакцій(приклад транзакцій) в реєстр обліку, відправивши ці транзакції до деяких вузлів, що беруть участь у блокчейні. Подані транзакції поширюються на інші вузли мережі (але це само по собі не включає транзакцію в блокчейн). Потім розподілені транзакції очікують у черзі або пулі транзакцій, доки вони не будуть додані до блокчейну.

Після створення кожний блок гешується, створюючи таким чином дайджест, що представляє блок. Зміна навіть одного біта в блоці повністю змінить геш-значення. Геш значення від блоку використовується для захисту блоку від змін, після того, як всі вузли матимуть копію геш-значення блоку, можна перевірити, чи не було змінено блок[10].

У блокчейні реєстр записів транзакцій є децентралізованим – це означає,

що одиночний комп'ютер або система не можуть отримати контроль над м реєстром. Завдяки ланцюгу транзакцій, що представляє собою довгий ланцюжок послідовних блоків, де кожен блок, що входить до цього ланцюга, є лише частиною загальної структури, яка бере свій початок від найпершої виробленої в даній системі операції стає неможливим будь кому змінити дані однієї транзакції, адже перед цим доведеться точно змінити всі наступні записи, що ведуть від цієї транзакції. Виходячи з цього, таке втручання виглядає вкрай складним процесом, що також є однією з переваг в побудові безпеки блокчейну.

Для підтвердження транзакції необхідно підтвердження більше двох користувачів, що забезпечують безпеку проведеної транзакції. Навіть в більшості сучасних систем в перевірці задіяні кілька рівнів верифікації: як правило, це продавець, покупець і певні «треті особи» (найчастіше банк або кредитне агентство).

Криптографічні ключі, які використовуються системою в обмінних процесах, також є важливим елементом сучасної кібербезпеки[12]. Кожен зашифрований ключ являє собою довгу, складну послідовність даних та практично не піддається розшифруванню без відомостей про приватний ключ. А якщо врахувати, що для підтвердження потрібні два таких унікальних ключа, то система починає задовольняти усім питанням безпеки. При цьому ще вдається зберегти майже повну прозорість транзакцій, що дає змогу перевірити правильність усіх даних.

Блокчейн, як і кожна система має потенційні вразливості, що можуть стати на заваді правильної роботи системи, наприклад, при створенні системи на основі технології блокчейн з нуля одна невелика помилка може стати фатальною. Команда проекту, що створюється на основі системи блокчейн або займається його розробкою та підтриманням дієздатності, має бути дуже досвідченою, бо вірогідність допустити помилку у такій складній системі підвищується.

Розміри мережі, що необхідні для роботи блокчейну – як мінімум кілька сотень узгоджено працюючих вузлів, через це система є вкрай вразливою до атак, коли, наприклад, якщо який-небудь користувач зможе отримати контроль над 51% вузлів системи, то він зможе повністю контролювати створення блоків у мережі, а коли в системі всього 20 вузлів, то подібний варіант розвитку подій

більш ніж можливий.

Структура блокчейну – при отриманні системою надто широкого поширення, інфраструктура блокчейну може виявиться не готовою до такого обсягу операцій, то в результаті може знизитися швидкість проведення транзакцій, можуть з'явитися проблеми зі зберіганням даних, що негативно вплине на ефективність мережі. В транзакціях в системі блокчейн використовуються публічні і приватні криптографічні ключі. Самі по собі такі ключі зламати майже неможливо на звичайному комп'ютері, проте зловмисник може отримати їх більш простим і звичним способом. Наприклад, ключі можна дістати в тому випадку, якщо ви зберігаєте їх на небезпечній або незахищеній платформі. Використання соціальної інженерії та викрадення ключів із менш захищених ресурсів є найбільш популярним способом на сьогоднішній день серед крипто валютних зловмисників [3].

2.5 Висновки за другий розділ

Було розглянено об'єкт інформаційної діяльності, що є транспортним засобом та використовує для захисту від несанкціонованого доступу систему безпеки високого класу з постійним моніторингом стану захищеності транспортного засобу за допомогою GSM зв'язку, використовуючи який дані відправляються на центральний сервер, до якого може отримати доступ власник для перегляду стану захищеності автомобіля у вигляді веб-сервісу. Розглянено захищеність компонентів об'єкту інформаційної діяльності, криптографічні протоколи, параметри та захищеність використовуваних геш-функцій, вибрано децентралізовану технологію, розглянуто її характеристики, проаналізовано попереднє використання в інших проектах, складові частини децентралізованого протоколу та можливості за використання децентралізованої технології в якості мережі передачі даних. Було вибрано децентралізовану мережу блокчейн, що повністю відповідає необхідностям побудови безпечної системи автентифікації для автомобіля.

3 ВИМОГИ ДО СИСТЕМИ АВДЕНТИФІКАЦІЇ

3.1 Вразливі місця у системі автентифікації автомобіля

Головне завдання сигналізації – сповіщати про будь-які неправомірні дії, скеровані на авто, і фактично кожна з існуючих моделей із цим справляється. Однак, техніка не стоїть на місці, і щойно зловмисники знаходять вразливі місця у захисних системах, як на ринку з’являються нові, покращені моделі. А значить, і сигналізація, яка ще пару місяців тому була доброю для Вашого сусіда/брата/свата, сьогодні може бути не достатньо доброю для Вас. Так само ненадійними, як дешеві або застарілі моделі, є і штатні заводські сигналізації, встановлені компаніями-виробниками авто. Адже їхній принцип роботи заздалегідь відомий крадіям, а стандартне місце розташування йде тільки їм.

Сьогодні на ринку є односторонні, двосторонні та GSM/GPS сигналізації. Найпоширеніші і водночас найпростіші автомобільні сигналізації – односторонні. Саме з таких моделей і почалось виробництво захисних пристроїв. Однак випробовування часом вони не пройшли і можуть використовуватись хіба в якості центрального замка. Адже варто лише віддалитись від автівки, як сигналу тривоги вже не чути і хоч якось запобігти викраденню стає неможливо.

Двосторонні та GSM сигналізації не лише включають сирену, але й відправляють відповідний сигнал на брелок або телефон власника і, крім того, наділені функціями віддаленого моніторингу авто. Додаткові можливості, які надають такі сигналізації включають також «інтелектуальне» керування світловими приборами, можливість блокування та запрограмованого запуску двигуна тощо. У той час як основним плюсом сигналізації з вбудованим GPS-приймачем є те, що навіть у випадку викрадення, поточне місце розташування авто можна визначити з точністю до п’яти метрів. Щоб уникнути такої ситуації, бажано, щоб охоронна система також була оснащена динамічним кодом сигналізації, замком

капоту з керуванням з додаткового каналу сигналізації та додатковим іммобілайзером на систему керування двигуном.

3.2 Модель загроз

Модель загроз на систему безпеки автомобіля включає в себе наступні атаки:

1) Втрата цілісності даних;

2) Підміна даних, атака типу «маскарад»;

3) Перехоплення важливих даних про роботу системи (дані про з’єднання, дані про власника, або навіть криптографічні параметри).

4) Атака типу DDoS;

5) Зламування криптоалгоритмів;

6) Пошук помилок у коді та «бекдорів»;

7) Соціальна інженерія.

Втрата цілісності і порушення даних сигналізації: система безпеки автомобіля повинна гарантувати, що всі дані, що обробляються в системі безпеки автомобіля, зберігаються та передаються, не були змінені будь-яким чином.

Загроза підміни даних з використанням атака типу «маскарад» для отримання доступу до мережі системи безпеки автомобіля або до даних користувач повинен пройти ідентифікацію, та на її основі пройти автентифікацію. Ідентифікація та аутентифікація виконуються на різних етапах процесу встановлення з'єднання або роз'єднання, при передачі даних до різних вузлів локальної мережі автомобіля, мережі близької дії для ідентифікації користувачів за допомогою мобільних пристроїв та передачі даних. При відсутності захисту від такого виду атак зловмисник може провести маскування себе як авторизованого користувача (атака типу «маскарад») і здійснити несанкціонований доступ (НСД) до компонентів системи безпеки.

DDoS (Distributed Denial of Service) – ще один тип атак, ідея якої полягає в пересиланні великої кількості схожих запитів на один конкретний сервер або цілу мережу серверів.

Зламування криптоалгоритмів. Наприклад, алгоритми для обчислення геш-функцій стандартів SHA-256 і ECDSA вважаються досить стійкими при існуючих обчислювальних потужностях. Однак, поява високопродуктивних квантових комп'ютерів збільшить ризик злому цих криптографічних функцій.

Щодо квантових комп'ютерів, є теорія, що вони в майбутньому перевершать за потужністю класичні системи. Але час появи цього моменту досить невідомий. Продуктивність квантових систем в певному діапазоні завдань вище, ніж у класичних комп'ютерів. Наприклад, система D-Wave, що використовує квантовий метод, в 100 мільйонів разів швидше звичайних комп'ютерів у завданні оптимізації.

Пошук помилок та «бекдорів» у коді протоколів. На сьогоднішній день це найбільша загроза для проектів, що побудовані на основі клієнт-серверної архітектури, оскільки написати повність безпечний код важко навіть для досвідчених розробників, то час від часу знаходяться нові помилки у коді, що можуть завдати суттєвої шкоди усій системі.

Соціальна інженерія. Цей спосіб є досить вдалим для зловмисників, оскільки багато людей не розуміють її принципи роботи систем безпеки автомобіля, що дає можливість маніпулювання довірою користувачів. Наприклад: існує багато прикладів, коли зловмисник переконував користувача перевести на його гаманець певну суму, а потім зникав. У випадку з автомобільними системами безпеки може бути використано довіру власника для отримання доступу до смартфону, ключам доступу чи просто міток доступу автомобіля. Також такі дії класифікуються законом, як крадіжка, та зловмисник понесе законне покарання.

3.3 Модель порушника

Модель порушника – абстрактне формалізоване або неформалізоване описання дій порушника, який відображає його практичні та теоретичні можливості, апріорні знання, час та місце дій тощо.

Стосовно систем безпеки автомобіля, порушники можуть бути зовнішніми або внутрішніми. Внутрішній порушник, у нашому випадку, – це, здебільшого, розробник певного проекту або додаткового компоненту (модуля) проекту, що хоче порушити безпеку системи ще на етапі розробки або експлуатації, реалізує внутрішнє проникнення або віддалене, де використовується вже відома помилка у безпеці. Зовнішній – атакує мережу, як звичайний користувач або учасник мережі системи безпеки автомобіля, реалізує віддалені атаки.

Мета порушника:

1) одержання можливості вносити зміни в роботу системи безпеки автомобіля згідно зі своїми намірами;

2) перешкоджати нормальній роботі мережі, що використовується для передачі даних (перевантажувати вузли, блокувати доступ, отримати контроль над мережею тощо);

3) отримання матеріальної або іншої вигоди, шляхом крадіжки даних автентифікації для доступу до автомобіля / провести викрадення автомобіля.

Для того, щоб здійснити атаку, внутрішній порушник повинен:

1) Мати доступ до редагування програмної частини мережі або створення та додавання до неї власних частин (бекдорів), що дадуть змогу використати систему безпеки у своїх цілях, для змінення параметрів автентифікації, додавання користувачів, підміну даних геолокації;

2) Володіти достатніми знаннями про роботу мережі та про її вразливі місця.

Зовнішній порушник повинен:

1) Володіти достатньою кількістю обчислювальних та матеріальних ресурсів для здійснення своєї атаки;

2) Володіти достатніми знаннями про роботу мережі та про її вразливі місця.

У ситуації для внутрішнього порушника, технічна оснащеність не є важливою тому, що використання бекдорів або заздалегідь створених каналів для несанкціонованого доступу є простою здачею для одного звичайного комп'ютера.

У ситуації для зовнішнього порушника, ресурси для атаки є вкрай важливими через те, що безпека багатьох сучасних блокчейн систем опирається на складність вирішення певних криптографічних задач. Успішність атаки залежить від кількості обчислювальних ресурсів / потужностей зловмисника, якщо мова не йде про використання знайденої помилки у мережі безпеки.

Кількість людей, що здійснюють атаку, може бути різною і залежить лише від плану самої атаки. Для внутрішнього зловмисника достатньо і однієї людини. Загалом, сучасні атаки на блокчейн або на користувачів систем безпеки автомобіля, здійснюються підготовленою командою із продуманим планом дій на кожному етапі атаки.

В нашому випадку при розгляді безпечності систем безпеки автомобіля, необхідно брати до уваги як внутрішніх, та і зовнішніх зловмисників. Основними вразливостями тут будуть канал передачі даних та сервер, що зберігає та обробляє дані користувачів.

3.4 Модель безпеки

Об’єкти системи безпеки автомобіля, що підлягають захисту:

1) Система зберігання даних автентифікації, що знаходяться в блоках автомобіля

2) Система зберігання даних автентифікації, що знаходяться на пристрої власника/того кому він делегує свої обов’язки

3) Дані автентифікації, дані власника, інформація про машину та її статут, що передаються за допомогою мережі інтернет

Засоби захисту інформації, що пропонуються:

Пропонується використання децентралізованої системи блокчейн на заміну клієнт-серверній архітектурі, що дозволяє:

1) Не використовувати сервер для зберігання даних про автомобіль, про стан його безпеки, геопозицію та даних авдетифікації, що дозволяє ускладнити доступ до перехоплення та модифікації даних на сервері при отриманні неправомірного доступу, оскільки такими серверами в системі блокчейн можна назвати ноди мережі, які й мають виконувати весь функціонал серверів з клієнт-серверної архітектури.

2) Важко відстежити ноди мережі щоб отримати до них неправомірний доступ та мати можливість впливати роботу системи з боку підтвердження невалідних дій від зловмисників та виконати DDoS атаку, бо дані про мережу не відомі.

3) Планується встановлена кількість нодів в мережі в кількості 10, що будуть працювати в будь-якому разі, оскільки будуть підтримуватися в робочому стані автоматичним менеджером нод, який має слідкувати за станом мережі та правильністю роботи системи, таким чином забезпечується постійний пул арбітруючих пристроїв, що повинно забезпечити безпечну роботи мережі блокчейн.

Захист об’єктів системи безпеки автомобіля за допомогою запропонованих засобів захисту інформації:

Криптографічний захист даних, що дозволяє вирішити проблему щодо атак на зміну цілісності даних, унеможливлює підміну даних та перехоплення важливих даних про роботу системи:

1) Шифрування даних, що зберігаються в блоці безпеки автомобіля

2) Шифрування даних, що передаються в мережі

3) Валідація даних, що передаються в мережі за допомогою геш-значень

4) Використання захищених каналів зв’язку

З використанням мережі блокчейн в якості мережі для передачі даних стає можливим використання механізмів керування доступом, що дозволяє

ефективно захищати інформацію, що курсує в системі безпеки автомобіля, даних ідентифікації, що передаються та їх верифікації за рахунок відсутності сервера, що може бути скомпрометованим та надійних засобів криптографічного захисту інформації.

В якості міток доступу використовується мобільний пристрій з достатнім рівнем довіри (в огляду на те, що криптоконтейнер, що пропонується використовується для зберігання даних банківських рахунків та іншої критичної інформації) з додатковим шифруванням блоковим симетричним «Калина», що є шифром державного стандарту та показує достатню стійкість для такої розробки.

3.5 Висновки за третій розділ

Були розглянуті можливості втрати цілісності і порушення даних в системі безпеки автомобіля повинна гарантувати, що всі дані, що обробляються в системі безпеки автомобіля, зберігаються та передаються, не були змінені будь-яким чином.

Для розуміння можливих атак були розглянені загрози, що влючають можливість підміни даних з використанням атака типу «маскарад» для отримання доступу до мережі системи безпеки автомобіля або до даних користувач повинен пройти ідентифікацію, та на її основі пройти автентифікацію, DDoS –, ідея якої полягає в пересиланні великої кількості схожих запитів на один конкретний сервер або цілу мережу серверів, зламування криптоалгоритмів, пошук помилок та «бекдорів» у коді протоколів що є найбільшою загрозою для проектів та соціальна інженерія, що є досить вдалим для зловмисників, оскільки багато людей не розуміють її принципи роботи систем безпеки автомобіля, що дає можливість маніпулювання довірою користувачів. Було створено модель загроз, модель порушника, модель загроз та модель безпеки для опису захищеності розробки.

4 РЕАЛІЗАЦІЯ СИСТЕМИ БЕЗПЕКИ

4.1 Вибір децентралізованої технології для передачі даних

Блокчейн – це розподілене цифрове сховище транзакцій, криптографічно захищені (підписані) і які є згрупованими у блоки. У свою чергу блоки є пов’язаними за допомогою криптографічної хеш-функції, тобто сформовані у вигляді одностороннього зв’язаного списку, де наступний блок вказує на попередній. За допомогою цього забезпечується захист від несанкціонованих змін у сховищі. Наступний блок повинен бути попередньо перевірений і підвергнутий розгляду за допомогою протоколу консенсусу. Таким чином чим більше нових блоків додається до сховища, тим складніше стає задача модифікації попередніх блоків. Оскільки кожний блок додається у розподілене сховище, безумовним є факт розповсюдження цього блоку в мережі серед учасників, і такий підхід призводить до конфліктів, коли до одного учасника приходять декілька блоків одночасно.

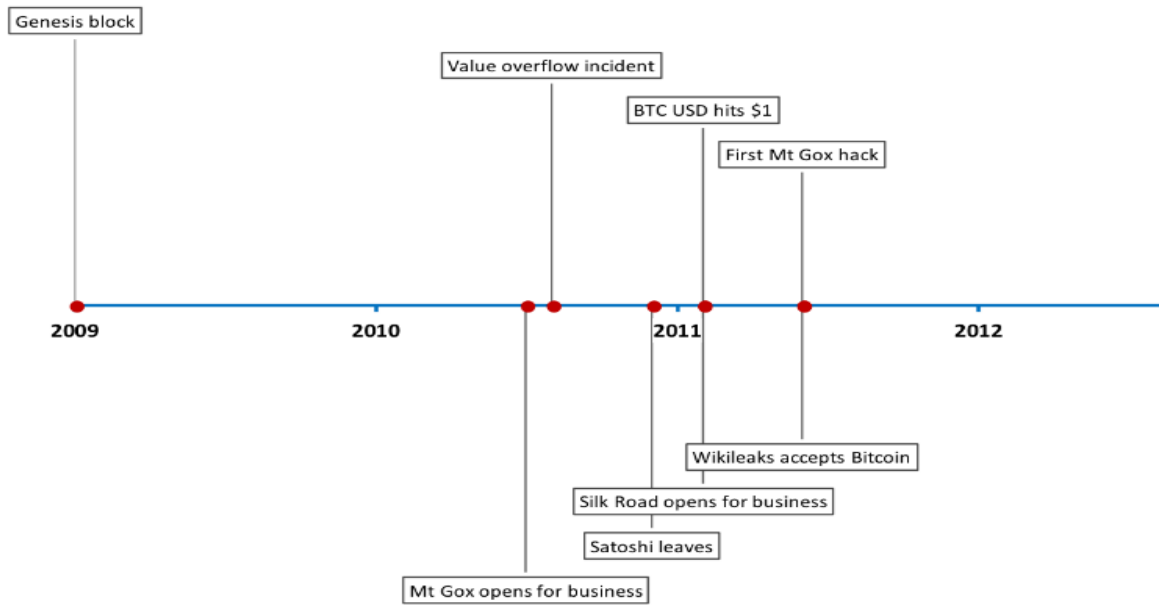

Рисунок 4.1 – Хронологія розвитку блокчейн

Початок розвитку блокчейну прийшовся на 2009 рік, коли і розпочалася історія блочейну і він розвивається і наразі(на рисунку 4.1 зображено хронологію розвитку блокчейну).

Згідно [11] основними характеристиками технології блокчейн є:

1) Захищеність – тобто технологія блокчейн повинна бути криптографічно захищеною;

2) Сховище – технологія використовує сховище, яке відкрите тільки для додавання нових записів для збереження повної історії транзакцій. Тобто записи які були колись зроблені не можуть бути перезаписані;

3) Розповсюдженність або децентралізованість – сховище поширене серед всіх учасників мережі;

4) Розподіленість – блокчейн може бути розподіленим, це дозволяє масштабувати кількість вузлів у мережі задля забезпечення більшої захищеності даних у сховищі.

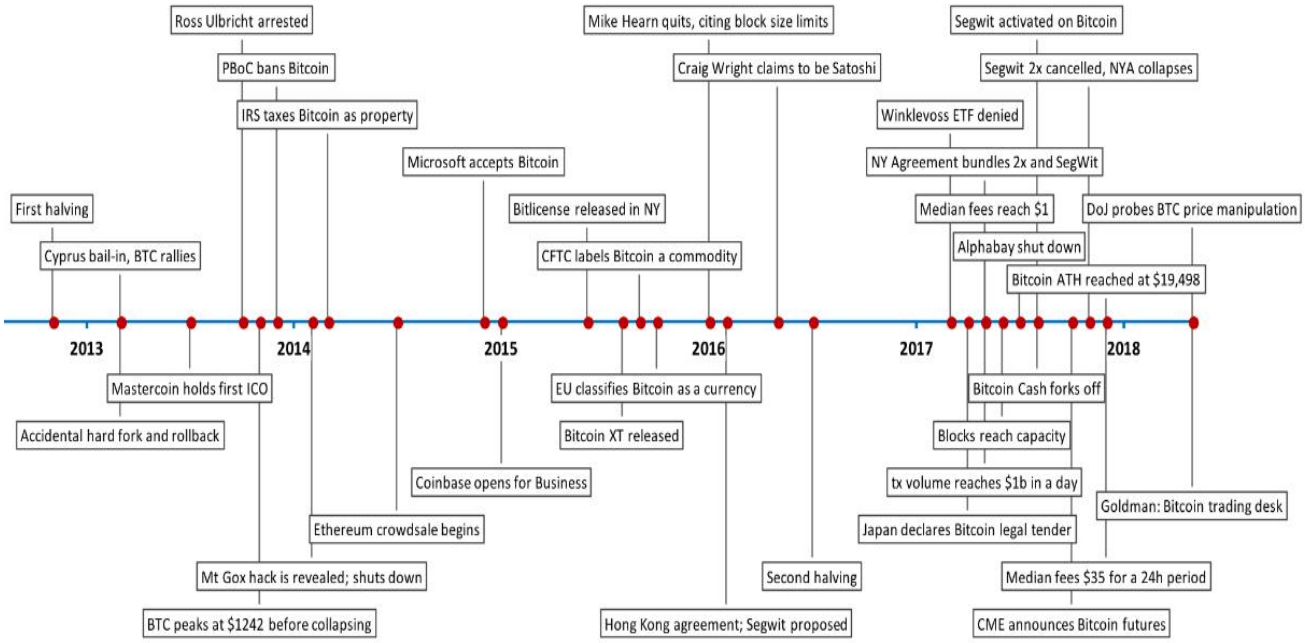

Рисунок 4.2 – Хронологія розвитку блокчейн (продовження)

Історично блокчейн як технологія може бути класифікована за відкритістю системи, але варто також відзначити, що класифікацію можна також вести за протоколами консенсусу, за стійкістю використовуваних алгоритмів та механізмів криптографічного захисту.

Відкритість блокчейну визначає можливості користувачів до його підтримки. Тобто, якщо підтримувати блокчейн можуть лише авторизовані користувачі (підтримувати, значить, наприклад, додавати блоки) такий блокчейн називається permissioned. Якщо ж підтримувати блокчейн може будь-який його користувач, така модель називається permissionless.

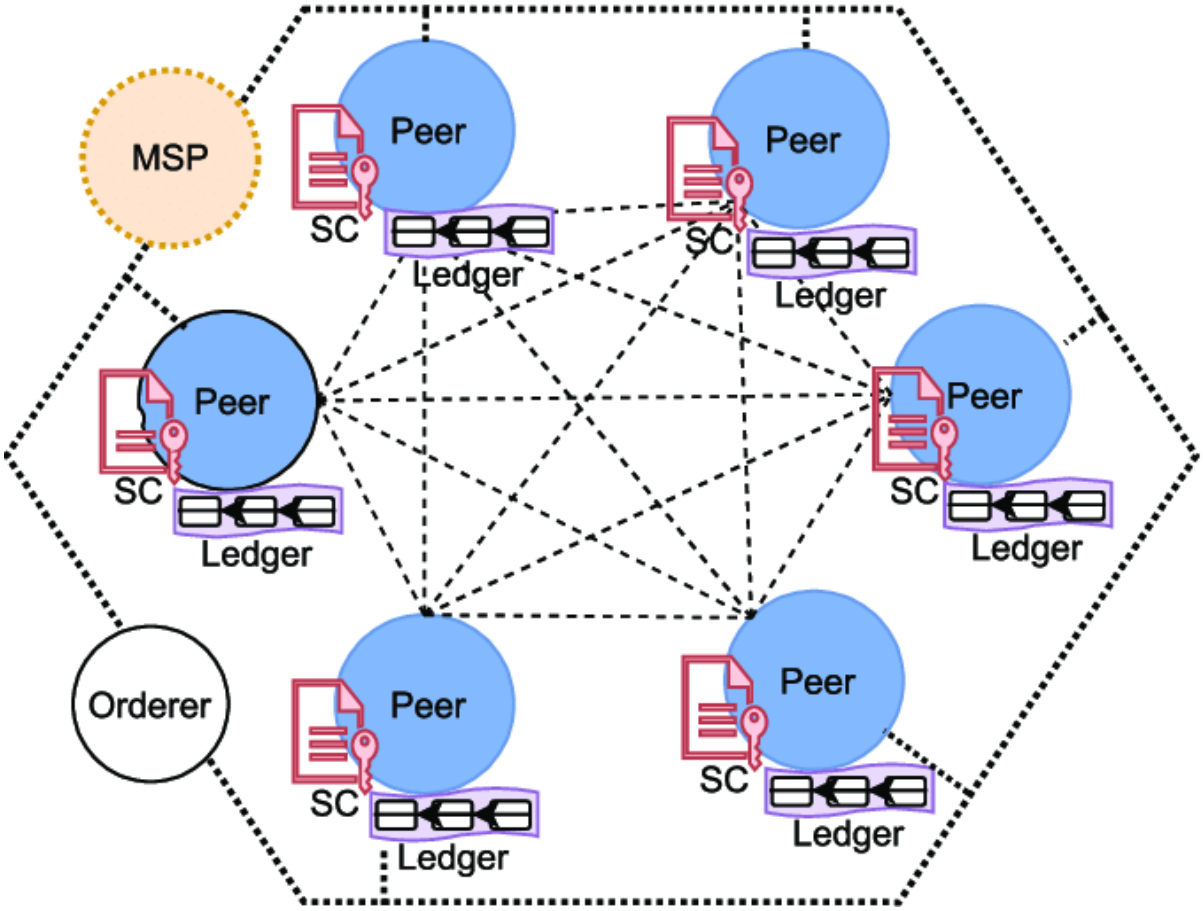

Рисунок 4.3 – Схема Permissioned – блокчейн

Permissioned – блокчейн(схема роботи зображена на рисунку 4.3), у якому доступ до його ресурсів та операцій над ними вимагає авторизації від деякого (централізованого або децентралізованого) органу. Попри цю особливість permissioned блокчейни можуть бути відкритим для читання будь-якому користувачу, або закритими. Найголовнішою особливістю є те, що запис до сховища

можуть робити лише обрані користувачі. Таким чином зменшується ймовірність атак або спотворення інформацій. Якщо все ж таки виникає спроба атаки, то користувача легко ідентифікувати і обмежити йому доступ до блокчейну.

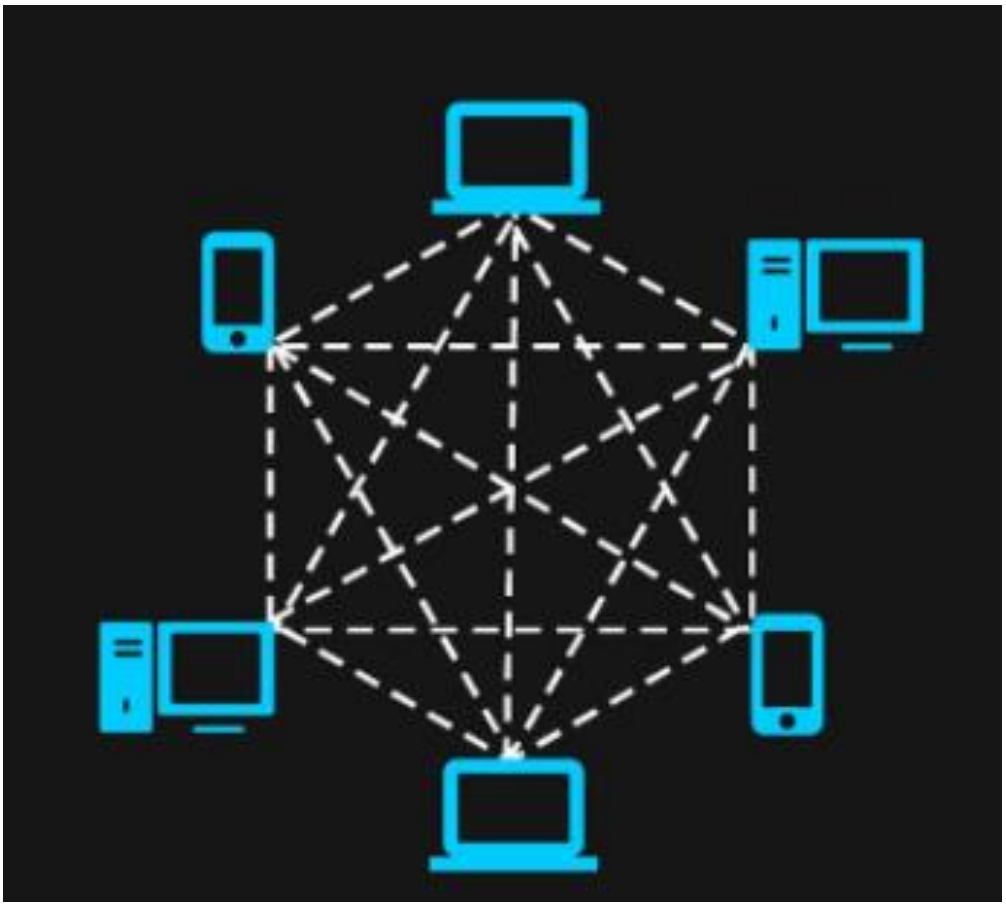

Рисунок 4.4 – Схема Permissionless – блокчейн

Permissionless – блокчейн(схема роботи зображена на рисунку 4.4), який є відкритим для читання та запису. Такі системи зазвичай є відкритими та будь-який користувач інтернету може стати його учасником. У випадку с permissionless блокчейнами, важливою відмінністю є те, запис може робити будь-який користувач, а отже збільшується кількість зловмисників. Через велику кількість користувачів серед яких можуть бути також і зловмисники виникає також необхідність у штучному елементі обмеження можливості модифікації сховища для користувачів. Такими елементами є протоколи консенсусу, сутність яких, зазвичай, полягає у тому, що користувачу необхідно витратити деякі ресурси для того, щоб додати блок у блокчейн.

Будь-який користувач який приєднується до мережі блокчейн автоматично погоджується з нинішнім станом системи:

1) Початковий стан системи;

2) Протокол консенсусу, який використовується у блокчейні;

3) Всі зміни, які були зроблені з часом (безумовно, при прийнятому протоколі консенсусу, вони можуть бути перевіреними користувачами незалежно);

Ключовою особливістю блокчейну є те, що, насправді, немає жодної необхідності у перевірці або поширенню стану системи третьою особою. Тобто, кожний користувач зберігає всю історію транзакцій та усі блоки і кожний користувач може власноруч перевірити справжність даних.

Проте для того, щоб встановити користувача, який має право на додавання наступного блоку даних необхідно обрати один з протоколів, який дозволив би однозначно і, здебільшого, випадковим чином обрати користувача. Такий підхід, власне, забезпечує гарантію того, що всі учасники будуть впливати на стан системи, и не буде одного учасника, який буде вирішувати які транзакції необхідно внести в даний блок, тобто таким чином гарантується децентралізованість мережі.

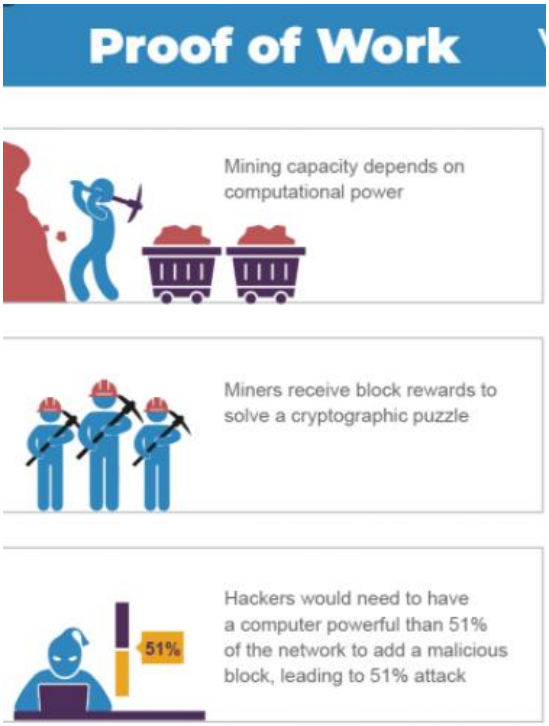

Рисунок 4.5 – Схема роботи підтвердження роботи в мережі блокчейн

Proof of Work – протокол консенсусу, який вимагає від користувача вирішити задачу, яка б вимагала значних обчислювальних ресурсів. Таким чином, для того, щоб додати блок до блокчейну користувачу потрібно довести, що він виконав певний, необхідний, об’єм роботи. Варто зазначити, що задача потрібна задовольняти певним умовам:

1) Будь-який користувач у певний момент часу може вирішити задачу (незалежно від обчислювальних ресурсів);

2) Складність пошуку рішення для задачі повинна мати експонтенційний характер, проте перевірка рішення повинне здійснюватися за поліноміальний час;

3) Задача повинна мати можливість регулювання за складністю, для того, щоб впливати на те, як часто відбуватиметеся публікація наступного блоку.

В основі своїй класичний блокчейн використовує два криптографічні примітиви: геш-функцію та електронний підпис. Геш-функція використовується для багатьох цілей:

1) Генерація адреси, або ідентифікатора користувача. В базовому розумінні система блокчейн використовує асиметричну криптографію, а саме електронний підпис для того, щоб ідентифікувати та підтвердити операцію або власника активу, який передає його іншому учаснику, або зміни документу. Звичайно, користувача можна ідентифікувати за його публічним ключем, проте публічний ключ в якості «пункту призначення» активу може бути завеликим за розміром, і не дуже сприйнятним для людини. Тому в якості адрес у системах блокчейн прийнято використовувати геш-значення від публічного ключа користувача;

2) Гешування транзакцій та блоку для підтвердження неспростовні блоку відносно помилок мережі а також геш-значення попереднього блоку використовується як посилання при формуванні наступного.

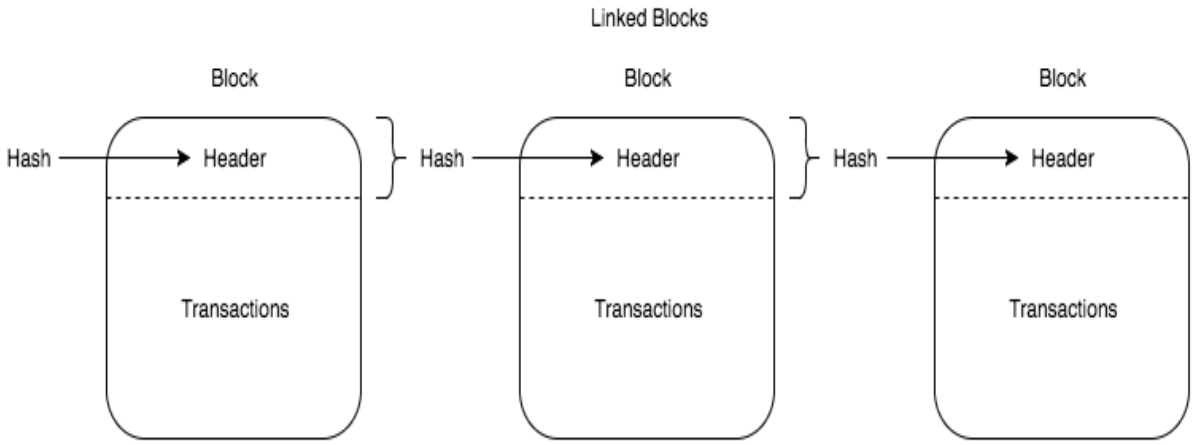

Рисунок 4.6 – Схема сховища даних блокчейну

У технологіях блокчейн використовується асиметрична ключова криптографія, що використовує пару ключів: публічний та приватний ключ, які математично пов'язані один з одним. Публічний ключ може бути оприлюднений без зниження безпеки процесу, в свою чергу приватний ключ повинен залишатись секретом, якщо інформація має зберігати свій криптографічний захист. Попри те, що існує взаємозв'язок між двома ключами, приватний ключ не може ефективно визначатися на підставі знання публічного ключа.

Криптографія асиметричного ключа використовує різні ключі від пари ключів для конкретних функцій, залежно від того, яка послуга повинна бути надана, наприклад під час цифрового підпису даних криптографічний алгоритм використовує приватний ключ для підписання. Підпис дозволяє перевірити що користувач, що передає значення іншому користувачеві, володіє приватним ключем, здатним підписувати значення за допомогою свого публічного ключа.

Користувачі можуть подавати кандидати транзакцій в реєстр обліку, відправивши ці транзакції до деяких вузлів, що беруть участь у блокчейні. Подані транзакції поширюються на інші вузли мережі (але це само по собі не включає

транзакцію в блокчейн). Потім розподілені транзакції очікують у черзі або пулі транзакцій, доки вони не будуть додані до блокчейну[3].

Після створення кожний блок гешується, створюючи таким чином дайджест, що представляє блок. Зміна навіть одного біта в блоці повністю змінить геш-значення. Геш значення від блоку використовується для захисту блоку від змін, після того, як всі вузли матимуть копію геш-значення блоку, можна перевірити, чи не було змінено блок.

Зверніть увагу, що коли користувач приєднується до системи блокчейн, користувач погоджується з початковим станом системи. Це записано в єдиний попередньо налаштований блок. Кожен блок блокчейну має опублікований блок налаштувальний блок та кожен блок повинен бути доданий до блокчейну після нього, на основі узгодженого методу консенсусу. Оскільки існує поєднання початкового стану та можливості перевірки кожного блоку, користувачі можуть погодитись з поточним станом блокчейну. Користувачі мають можливість перевіряти кожен блок. З кожним роком кількість сфер застосування і новітніх розробок на основі технології блокчейн зростає. Децентралізована система зберігання електронних даних набула широкого використання із самого початку у криптовалютах.

Оскільки зберігання даних для систем автентифікації(в тому числі для систем автентифікації автомобіля), вимагає надійної системи зберігання даних, розглянемо можливості блокчейна з зберігання даних.

База даних блокчейн складається з послідовності транзакцій, інформація про які зберігається одночасно на великій кількості комп'ютерів в різних точках світу.

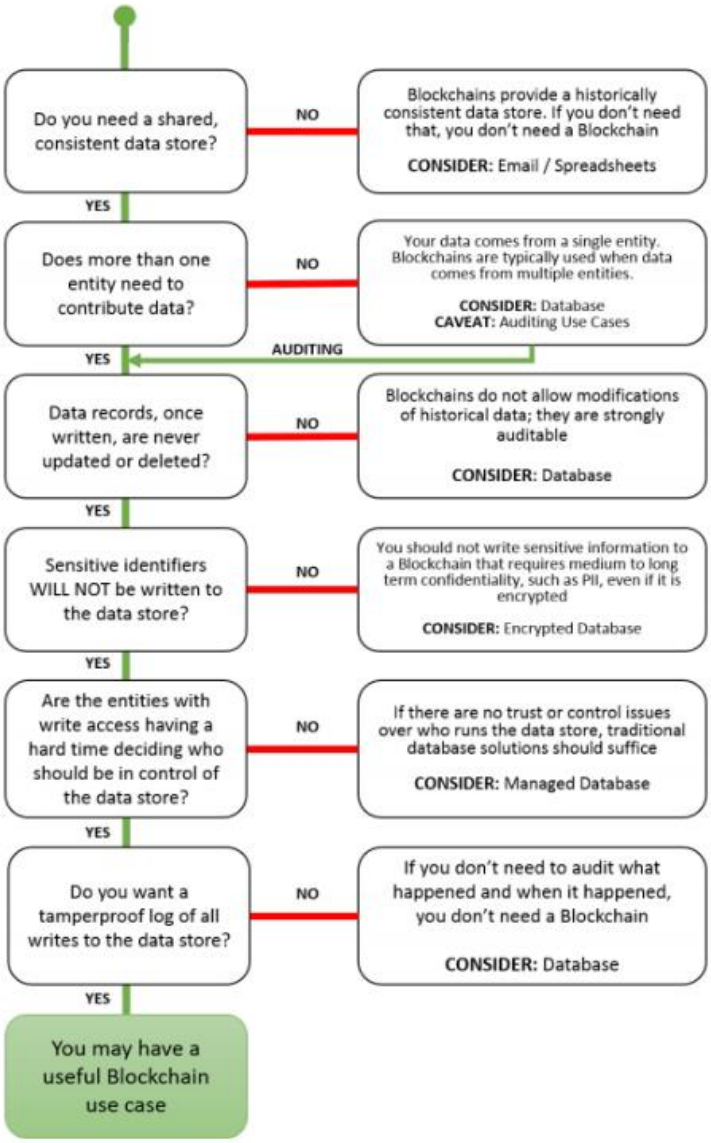

Рисунок 4.7 – Рекомендації щодо вибору системи сховища даних

Підробити або видалити такі відомості нереально або дуже складно: саме ця особливість робить блокчейн технологію такою перспективною для використання в медицині, логістиці, управлінні даними, державних і багатьох інших сферах регулювання. Розглянемо основні з них:

1) Логістика. Майже кожен етап перевезення і доставки вантажів можна відстежити за допомогою блокчейн-технологій. Логістичні системи такого типу дозволять зрозуміти, де знаходиться замовлення в дану хвилину, в яких умовах зберігається, хто відповідає за його перевезення у даний момент. Адміністрування мереж та підтвердження прав доступу.

2) Паролі, імена та кількість користувачів певної мережі можуть бути надійно захищені за допомогою технології блокчейн. При відсутності одного головного центру адміністрування таку базу стає неможливо зламати – вона захищена від хакерів і серверних помилок, бо завжди буде її легітимна копія у іншого користувача, яку можна почати використовувати в будь-який момент.

3) Дані про право власності. Технологія блокчейн легко адаптується для виконання будь-яких операцій по реєстрації, передачі та зберігання даних. З її допомогою можна підтверджувати і передавати інформацію про права власності за хвилини: вкрасти або підробити такі дані дуже складно.

4) Створення безпечної системи DNS. Застосування блокчейну в службі доменних імен допоможе захиститися від DDoS-атак. Децентралізована альтернатива HTTP для реєстрації доменних імен не дозволить хакерам заблокувати роботу відразу всієї системи і ефективно атакувати сервери.

Мережа блокчейн дозволяє безпечно зберігати критичні данні користувачів, розглянемо схему рекомендації вибору системи сховища даних(зображено на рисунку 4.7), з якої бачимо, що мережа блокчейн гарно підходить в якості системи сховища критичних даних системи.

4.2 Огляд складових ІС автомобіля, що підлягають захисту

В загальному вигляді інформаційна система, що розглядається складається в загальному розумінні з кількох підсистем, що взаємопов’язані між собою та виконують свої функції, що в сукупності і дозволяє компонентам системи синхронізовано та правильно між собою.

Підсистема обробки інформації забезпечує створення, зберігання, актуалізацію інформації про стан блоків транспортного засобу та можливості керування ними і складається із засобів обробки інформації, системного та функціонального ПЗ.

До засобів обробки інформації належать бортовий комп’ютер та встановлена кількість блоків керування для забезпечення всіх функцій щодо роботи,

супроводження транспортного засобу та захисту інформації.

Підсистема взаємодії з користувачами АС забезпечує моніторинг, керування даними блоків транспортного засобу, з використанням мереж передачі даних та стандартних CAN та LIN-протоколів.

Технічні засоби, що встановлені на даному транспортному засобі та цікавлять нас при проведенні даного типу огляду взаємопов’язані в мережу CAN, або контактують з нею. Мережа CAN ділиться на два рівні абстракції:

CAN High, в колі якого взаємодіють надзвичайно важливі для правильної роботи механізмів транспортного засобу блоки, наприклад блок керування двигуном (рисунок 4.8) або блок керування автоматичною трансмісією, компрометація яких може призвести до трагічних наслідків та CAN Low,

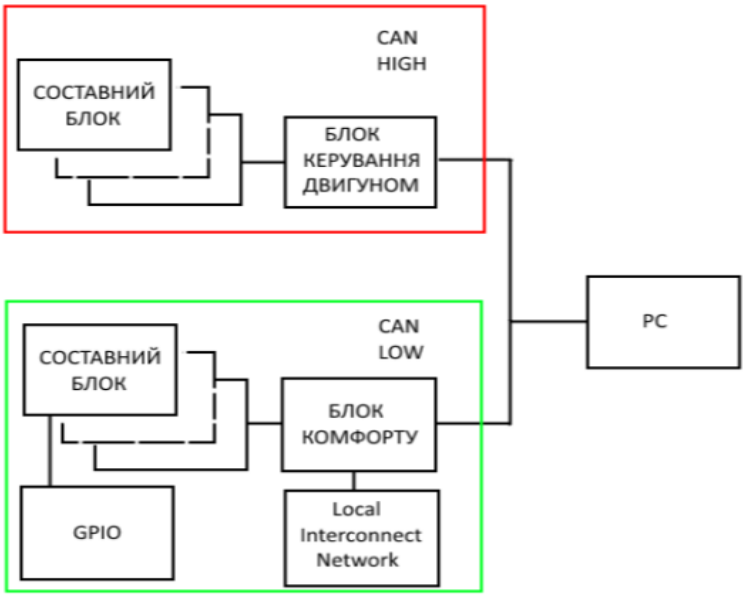

Рисунок 4.8 – Склад робочої станції

в якому взаємодіють блоки, компрометація яких не є такою критичною, наприклад блок керування комфортом чи яким-небудь мультимедійним пристроєм.

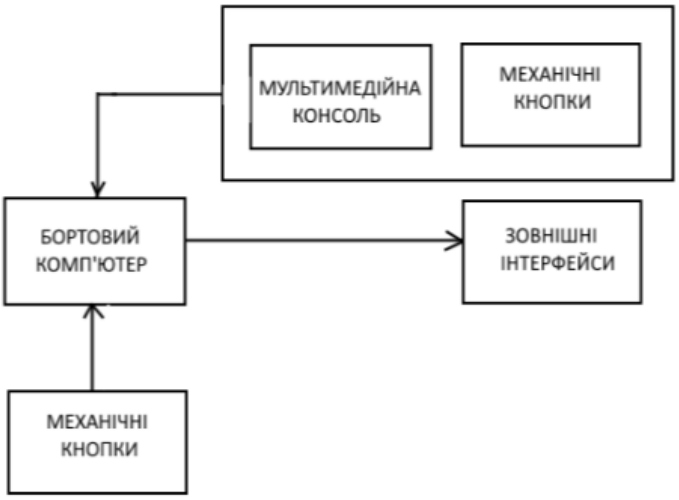

Рисунок 4.9 – Взаємопов’язані з PC компоненти

Таке розподілення є додатковим видом абстракції над HAL (Hardware Abstraction Layer), що ускладнює доступ до більш високого рівня абстракції через більш малий рівень, це зроблено для того щоб при тому, що пристрій, який буде контактувати напряму до мультимедійної системи, або через інші інтерфейси, через які може бути проведена атака на цей блок не буде використаний для атаки на більш важливі блоки, які знаходяться за рівнем абстракції.

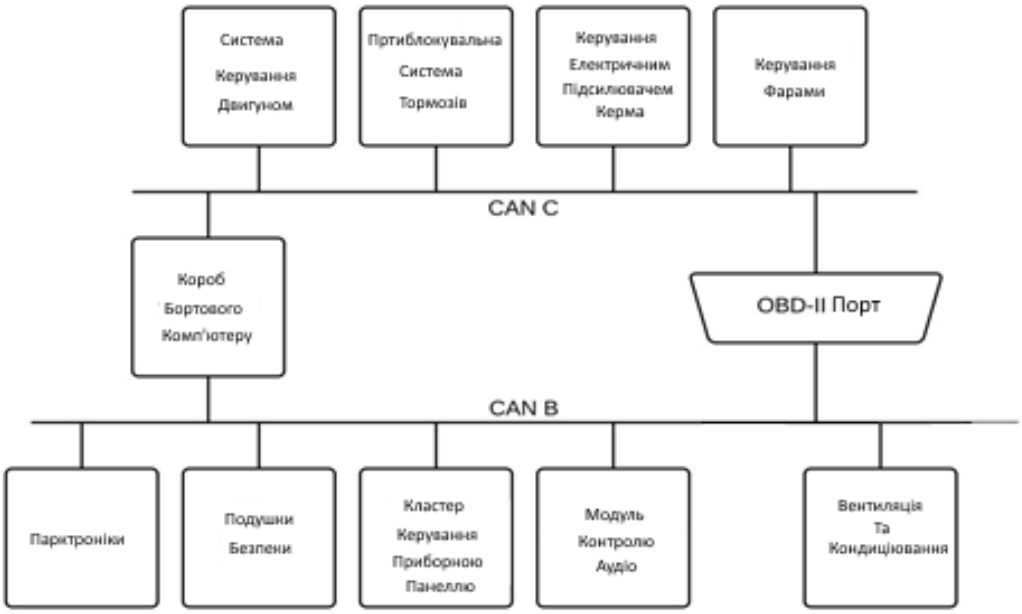

Рисунок 4.10 – Розподілення блоків за рівнями абстракції

В рисунку 4.10 показано як розподіллено рівні абстракції мережі CAN, яка підтверджує ствердження вище, на рівні CAN C знаходяться блоки керування двигуном, блоком ABS, електронного посилювача руля та контролю фар, а на рівні CAN B – керування парктроніками, керування подушками безпеки, аудіо системою, системою кондиціонування. Мережа CAN має в своєму складі велику кількість електронних блоків керування, моніторингу, нас цікавлять ті що взаємопов'язані між собою мережею CAN.

Також транспортний засіб підтримує високошвидкісний тип зв'язку за допомогою з'єднань через оптоволокно, що може бути налагоджена через діагностичний інтерфейс.

Розглянемо для наочності структурну схему з'єднання компонентів в мережі CAN, в якій виділимо основні блоки компрометація/захоплення доступу яких може складати зацікавленість.

Таблиця 4.1. – Електронні блоки, пов'язані мережею CAN

Блок	Опис
A1	Приладова панель
CAN B	Шина даних CAN рівню абстракції B
CAN C	Шина даних CAN рівню абстракції C
N15/3 (H 15/3)	Блок управління ETC
N15/5(H15/5)	Блок управління важелем селектора AT
N47-5(H47-5)	Блок управління ESP, SPS и BAS
N2/7(H2/7)	Блок управління систем самодіагностики
K12	Електронний стартовий вимикач запалювання (EZS)
CAN	Шина даних
N22(H22)	Мультимедійна панель
K15	Система Parktronic (PTS)
K23	Подушки безпеки з вбудованою системою виклику ARMINCA
CAN C	Привід и ходова частина
K26	Електронний блок селектора передач (EMW)
K28	Електронна протизаносна система (ESP)
N10/1(H10/1)	Блок комфорту
N10/2(H10/2)	Блок заднього SAM

В першу чергу це блоки N3/9 (блок управління *CDI* (Capacitor Discharge Ignition) (електронне запалювання)), N22 – блок при бортової панелі, центральний інтерфейс N93, K26, K28, N2/7 та найбільшу зацікавленість складає блок N22 (додаткове пояснення скорочень назв блоків представлено у таблиці 4.1).

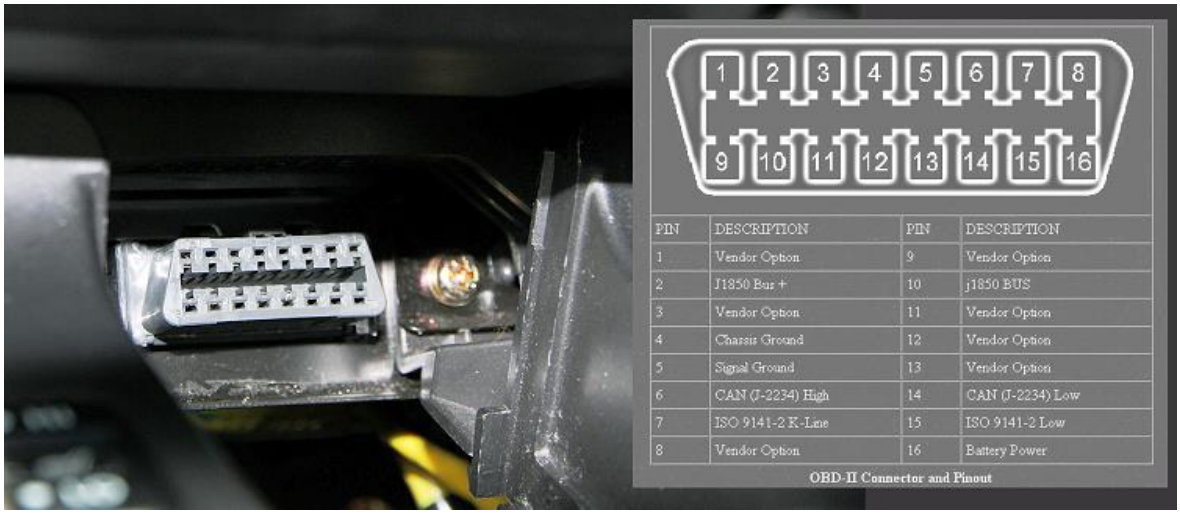

Рисунок 4.11 – Інтерфейс OBD-II

Діагностичним інтерфейсом виступає OBD-II(зображений на рисунку 4.11), який дозволяє підключити пристрій, який має такий інтерфейс, та на якому встановлено необхідне програмне забезпечення для моніторингу стану пристроїв, що пов’язані з мережею CAN, їхнім керуванням та налаштуванням.

Мережа CAN розподілена в даній реалізації на 3 рівні безпеки (доступу) (A, B, C), в яких знаходяться блоки відповідно за рівнем важливості. Це зроблено для того, щоб пристрої які мали менший рівень не змогли б впливати напряму на роботу інших блоків, наприклад в більш ранніх версіях такого типу мережі була можливість проведення атаки через компрометацію наприклад блоку K12 з інтеграцією спеціалізованого пристрою з радіокеруванням в загальну шину мережі CAN і таким чином можна було вимкнути заживання шляхом відправки команди на блок K12, який вимикав двигун транспортного засобу.

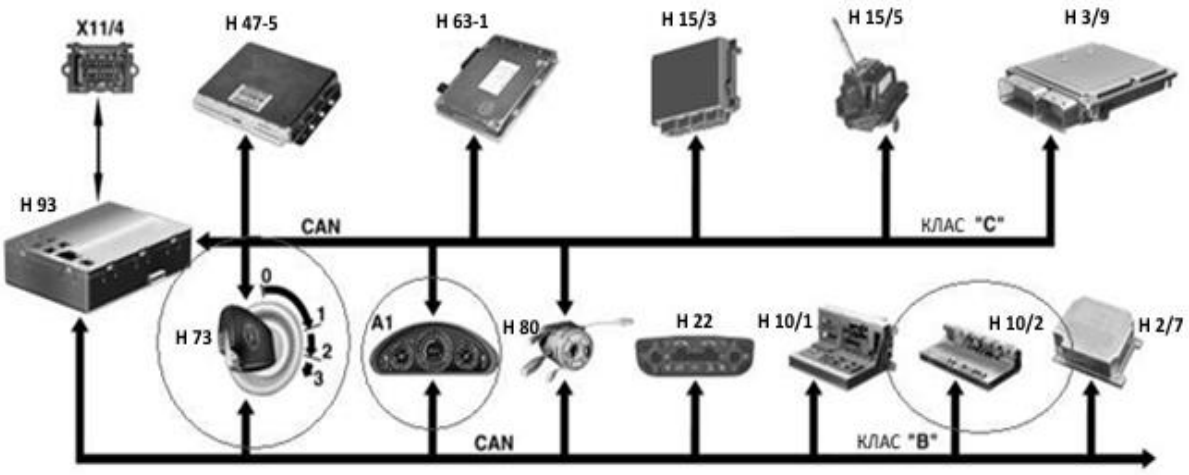

Рисунок 4.12 – Схема з’єднання блоків між собою в мережі CAN

Де РС – робоча станція, CAN К – комутатор мережі, БЛК Х.Х/х – блок відповідної номенклатури.

Багаторівнева система дозволяє не допустити атаку такого типу напряму за будь-якої частини підключення до CAN мережі (High, Low).

Найчастіше атаки проти компонентів CAN направлені саме на компрометацію блоків, але є приклади атак направлених на збір інформації з різних блоків для майбутнього використання для злому безпеки авто чи розробки інших атак у майбутньому.

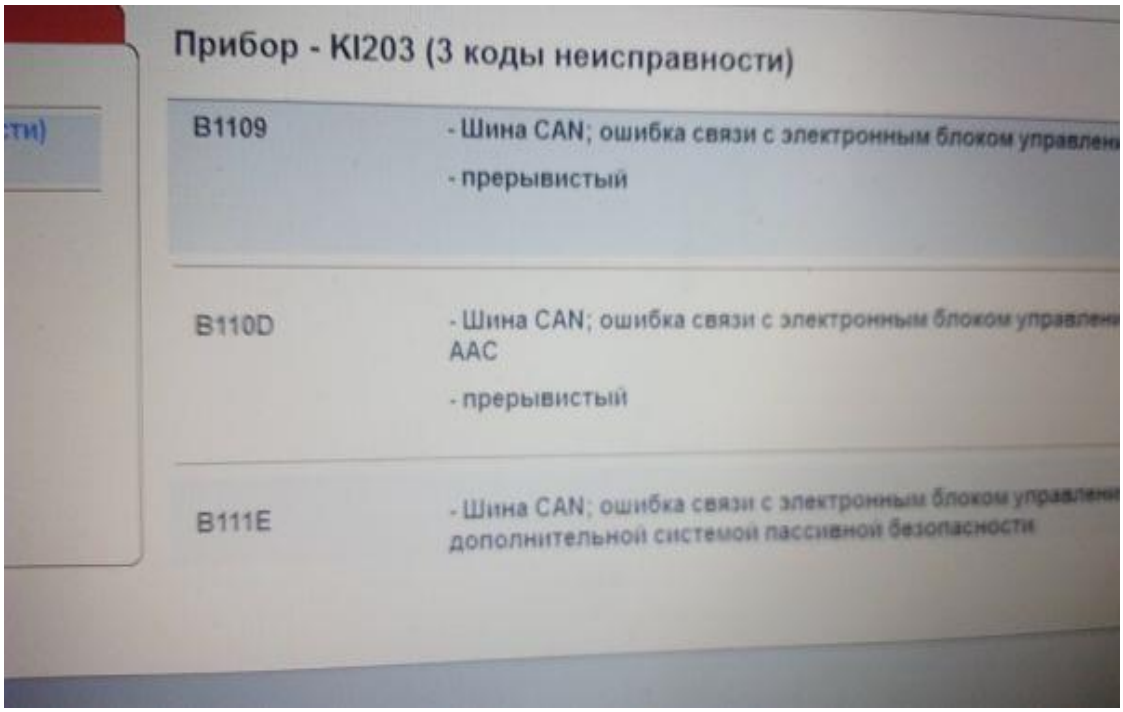

Рисунок 4.13 – Зчитування помилок в мережі CAN

Такий інтерфейс пов’язує всі електричні компоненти автомобіля та дозволяє проводити діагностику вузлів автомобіля, використовуючи дані з пристроїв, дані про обриви, замикання, результат системи моніторингу показано на рисунку 4.13.

Найпростіший прикладами датчиків є пристрої бінарного типу, швидкозмінні, датчики рівня.

Вони доступні користувачу транспортного засобу, та мають вигляд піктограм на панелі приборів автомобіля, поряд в таблиці вказано параметри на даний момент часу, що використовує сама панель приборів.

Цю інформацію можна отримати в бінарному вигляді з використанням підключення до мережі CAN подальшим переведенням даних у більш простий для читання вид.

Перевівши дані в байти інформації можна відстежувати зміну роботи вузів та агрегатів, написавши простий скрипт або програму можна отримувати дані на вхід, аналізувати ці дані за певний проміжок часу та видавати користувачу дані про стан транспортного засобу.

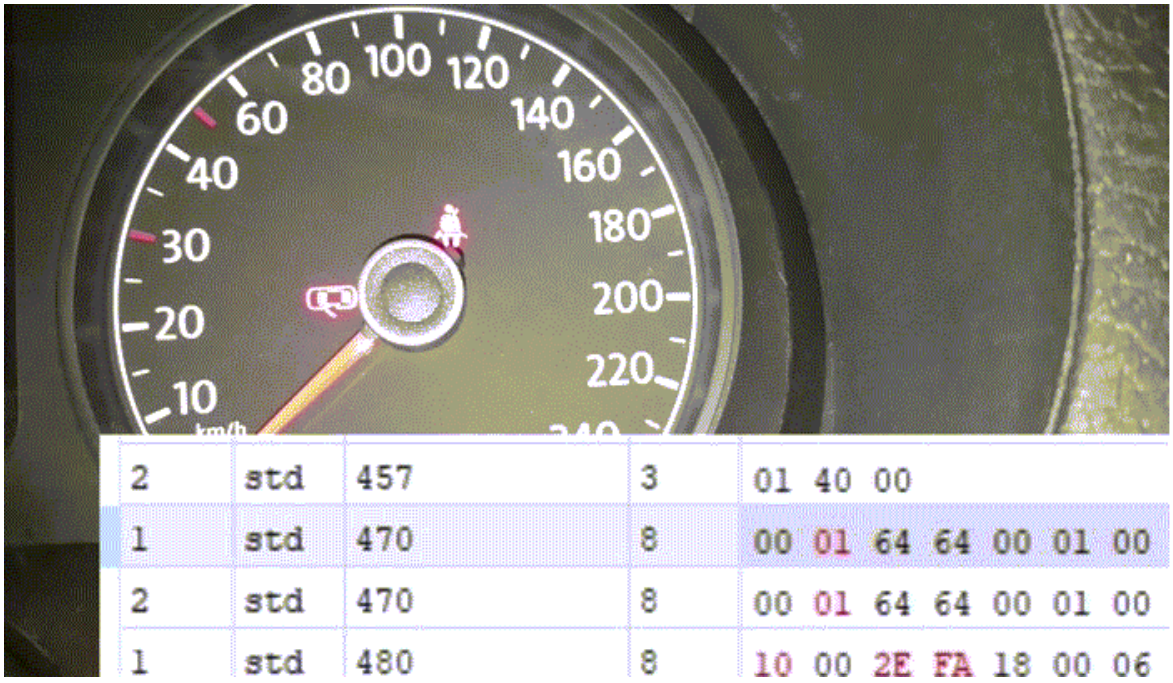

Рисунок 4.14 – Робота панелі приборів автомобіля

Для прикладу, розглянемо програму для моніторингу стану рівня палива у автомобілі.

Рисунок 4.15 – Програма-монітор кількості пального

Програма використовує платформу андроїд, до якої через бездротову мережу підключено адаптер для зв’язку з мережею CAN.

Таблиця 4.2 – Приклад параметрів що передають блоки в CAN

NAME_DESC	FACTOR	UNIT	SID	LID	BYTE_POSITION	BYTE_LENGTH
PRES_Fact_SWORD	0,00012207	-	0x21	0x6B	24	2
PRES_Sig_Fact_CSTR_1b	0,008	-	0xFF	0xFF	13	1
PRES_Sig_Fact_CSTR_1b	0,008	-	0xFF	0xFF	27	1
PRES_lambda_CARB_UWORD	3,05E-05	-	0x21	0x6B	34	2
PRES_lambda_SWORD	0,001	-	0x21	0x6B	32	2
PRES_OneToOne_ULONG	1	-	0x21	0x6C	2	4
PRES_OneToOne_ULONG	1	-	0x21	0x6C	6	4
PRES_Period_ms_SWORD	0,1	ms	0x21	0x2A	18	2
PRES_AirMassPerTime_SWORD	0,1	kg/h	0x21	0x21	24	2
PRES_OneToOne_UBYTE	1	-	0x21	0x67	31	1
PRES_Time_PwmHigh_UWORD	0,03571428	us	0x21	0x22	21	2
PRES_Pwr_SWORD	9,424778	W	0x21	0x21	12	2
PRES_Pwr_SWORD	9,424778	W	0x21	0x29	16	2
PRES_Pwr_SWORD	9,424778	W	0x21	0x32	16	2
PRES_Pwr_SWORD	9,424778	W	0x21	0x25	21	2
PRES_Pwr_SWORD	9,424778	W	0x21	0x44	6	2
PRES_Pwr_SWORD	9,424778	W	0x21	0x81	19	2
PRES_Pwr_SWORD	9,424778	W	0x21	0x83	27	2

Зовнішній вигляд програми наведений на рисунку 4.15, де видно, що система визначає рівень палива в підключеному автомобілі через мережу CAN, приклад параметрів, що отримуються таким способом зазначено в таблиці 4.2.

Розглянемо пристрій в мережі, що працює в стандартному режимі та у випадку, коли до мережі підключено скомпрометований пристрій.

Рисунок 4.16 – Робота в стандартному режимі

В загальному вигляді пристрій, навіть якщо він не скомпрометований буде підключений так, як показано на рисунку 4.12, тому для мережі при правильному налаштуванні компрометуючого пристрою (вірно проставлених ідентифікаторів пристрою ітд) буде виглядати як нескомпрометований пристрій, що готовий до роботи.

На рисунку 4.16 показано з’єднання пристроїв у стандартному режимі, коли в мережі немає скомпрометованого пристрою та повідомлення передаються вірно від відправника до отримувача.

В наступному випадку введемо можливість компрометації пристрою, це буде зроблено в декілька етапів. По перше, треба вибрати пристрій для компрометації, наприклад розглянемо як такий пристрій ECU2, що є яким-небудь легкодоступним пристроем, визначимо, що ECU2 буде видавати себе за пристрій ECU4, який наприклад буде пов'язаний з ECU двигуна. Для того щоб ECU2(NEW ECU4) став повністю працюючим скомпрометованим пристроєм необхідно щоб пристрій не був дублікатом, тому необхідно вимкнути нескомпрометований пристрій ECU4. Для того щоб вимкнути пристрій з мережі можна використати DOS атаку. Данні атаки мають багато варіантів реалізації, розглянемо деякі з них:

Наприклад, DoS всієї шини. Дозволяє повністю запобігти комунікації CAN, генеруючи в шині безперервні з’єднання, цей стан не дозволить будь-якому вузлу надсилати повідомлення.

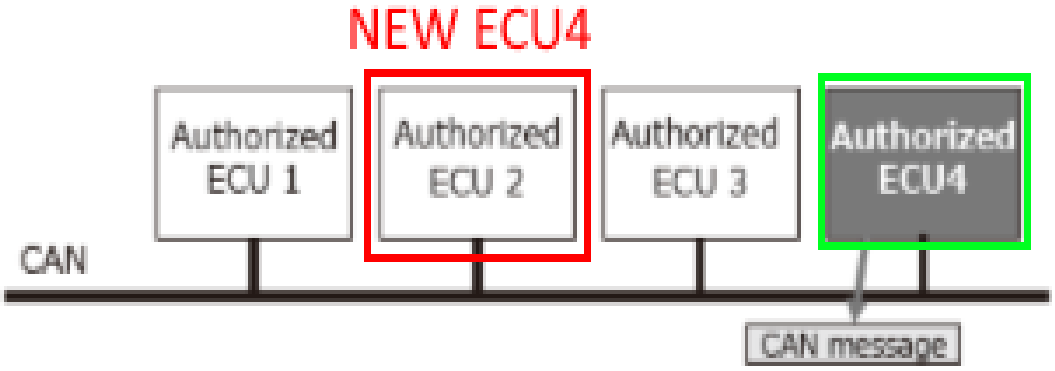

Рисунок 4.17 – Вибраний пристрій для компрометації

Для проведення атаки, необхідно згенерувати постійний рівень «0» в лінії приймача CAN. Багато реалізацій CAN мають вбудовані механізми запобігання таким порушенням. Але цей варіант не підходить для даного виду атаки, оскільки так вся мережа стане не доступною

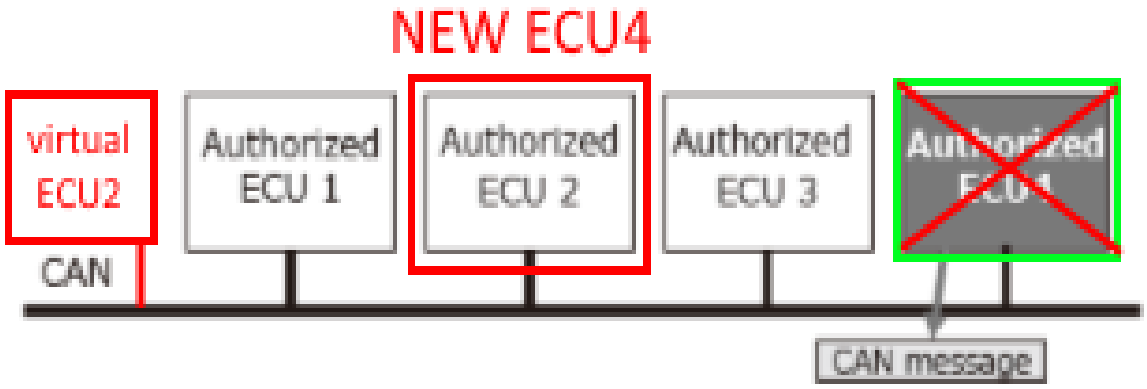

Рисунок 4.18 – Вибраний пристрій для компрометації

Направлена DoS атака. DoS ACK може бути направлена на один пристрій в мережі, вводячи домінуючі біти тільки в повідомленнях, надісланих на цільовий вузол. Для цього атакуючий повинен мати точні дані про ідентифікатори повідомлень, відправлених певним вузлом. Контролюючи шину для цільових ідентифікаторів, вона повинна вводити необхідні біти після фази арбітражу. Така заміна призводять до втрати цільових даних, лічільник помилок зростає, мережа

видаляє пристрій з пріоритетності передачі і тим самим пристрій можна вважати вимкненим. Цей варіант повністю підходить для даного виду атаки, оскільки в цьому випадку стає недоступним лише пристрій, що атакується, а вся мережа працює в звичайному режимі.

Протестувати роботу концепту з компрометованим пристроєм можна за допомогою спеціального симулятора автомобіля, який дозволяє детальніше вивчити роботу CAN мережі з різними пристроями.

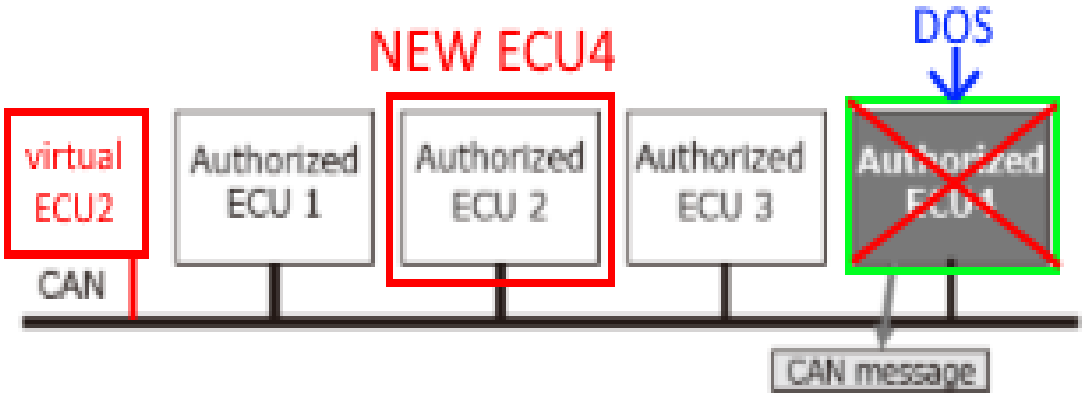

Рисунок 4.19 – Робота в стандартному режимі

В налаштуваннях пристрою(одного з ECU) змінимо параметри пристрою на інші. Детальніше програмування ECU буде розглянемо в наступному розділі. Після тестування даної атаки визначено, що для мережі та пов’язаних пристроїв новий пристрій ECU2, що наслідував усі параметри пристрою ECU4 повністю замінив його в мережі, що атакувалася.

В якості прикладу такої атаки можна привести нещодавно знайдена вразливість у сучасних електрокарах класу «люкс»,
що може викликати великі проблеми роботи системи безпеки, наприклад можуть бути скомпрометовані блоки автомобіля та угон транспортного засобу.

Tesla Model X, що використовувала систему безключового доступу для встановлення необхідних оновлень, що має в собі помилки, що дозволяє угнати авто за 5-10 хвилин[10].

Рисунок 4.20 – Зовнішній вигляд ключа Tesla Model X

Дослідник виявив як в системі безключового доступу Tesla Model X так і в самому автомобілі уразливості, що дозволили йому через Bluetooth-підключення переписати прошивку брелока(зовнішній вигляд показаний на рисунку 4.20), зняти код розблокування і викрасти автомобіль.

За словами дослідника, викрадач, якому вдалося прочитати ідентифікаційний автомобіля номер (зазвичай його видно на приладовій панелі автомобіля через лобове скло) і наблизитися до брелоку жертви на відстань 4,6 м, зможе проексплуатувати ці уразливості. Необхідна для цього обладнання обійдеться в $300, його спокійно можна вмістити в рюкзак, а управління здійснюється за допомогою смартфона.

Рисунок 4.21 – Зовнішній вигляд Tesla Model X

Всього за 90 секунд представлене Ваутерсе пристрій здатний витягти радіокод для розблокування Tesla Model X. Опинившись всередині автомобіля, викрадач може проексплуатувати другу уразливість і всього за хвилину завести машину за допомогою власного брелока.

4.3 Підтвердження концепту архітектури системи безпеки

Основу концепту системи безпеки складає система автентифікації, що в якості системи передачі даних використовує децентралізовану систему блокчейн.

Розглянемо детальніше вплив загроз на складові архітектури на основі рисунку 4.21. З лівої сторони показано передачу пакетів з даними не використовуючи мережу блокчейн, а просту клієнт-серверну архітектуру, де можна побачити

першу основну проблему: пакети передаються у відомому порядку, відомий отримувач та відправник, що дозволяє зловмисникам виконувати атаки з перехоплення, підміни, перенаправлення, повторної передачі на такого роду мережу.

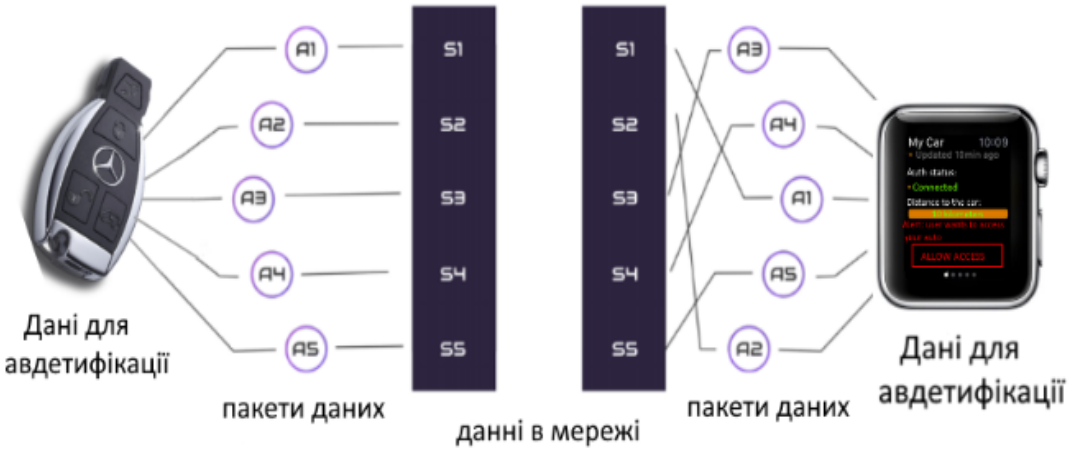

Рисунок 4.22 – Порівняння зберігання даних в мережах без та з використанням мережі блокчейн

Також дані, що передаються не завжди гарно захищені з огляду на криптографічних захист інформації.

Можливо перехоплення даних також у локальній мережі, оскільки використовується той самий підхід, тільки дані передаються в локальній мережі поряд з автомобілем та не мають гарної захищеності через пропускні можливості такого каналу зв’язку, що є дуже критичною проблемою.

Порівняння зберігання даних в мережах без та з використанням мережі блокчейн (зображено на рисунку 4.22 з правої сторони), в якому вирішена проблема того, що маршрути пакетів відомі, оскільки використано засоби децентралізованого з’єднання між учасниками мережі. Другою проблемою, було те, що локальні засоби автентифікації не дозволяють використовувати гарний криптографічний захист через вузький пропускний канал та малі можливості з обчислення(беручи до огляду системи безпеки, що не використовують мережу інтернет для передачі даних), в даній архітектурі використовується з’єднання до мережі інтернет і може бути використаний гарний криптографічний захист.

Таким чином, запропонована архітектура, що побудована на основі технології блокчейн показує більш гарну захищеність в порівнянні з простою клієнт-серверною архітектурою.

4.4 Програмна реалізація моделювання роботи системи безпеки

Під час роботи системи додаються ноди системи безпеки, які являють собою пристрої автовласників, що обмінюються між собою повідомленнями, які складають блоки даних. Розглянемо детальніше структуру блоків: вид даних(наприклад: пакет автентифікації), сервісний код(унікальний ідентифікатор для ідентифікації транспортного засобу), код повідомлення(унікальний код для збереження черг повідомлення) та саме повідомлення, що має в собі інформацію, яку необхідно валідувати в мережі блокчейн.

Для моделювання роботи системи безпеки було створено програмну модель, що складається модулю ініціальної реєстрації транспортного засобу в системі, модулю перевірки та підтвердження даних користувачів, модулю зберігання ключових даних користувача та комплексу систем моделювання атак на розроблену систему, що дозволяє протестувати запущену та сконфігуровану систему на реакцію при моделюванні атак.

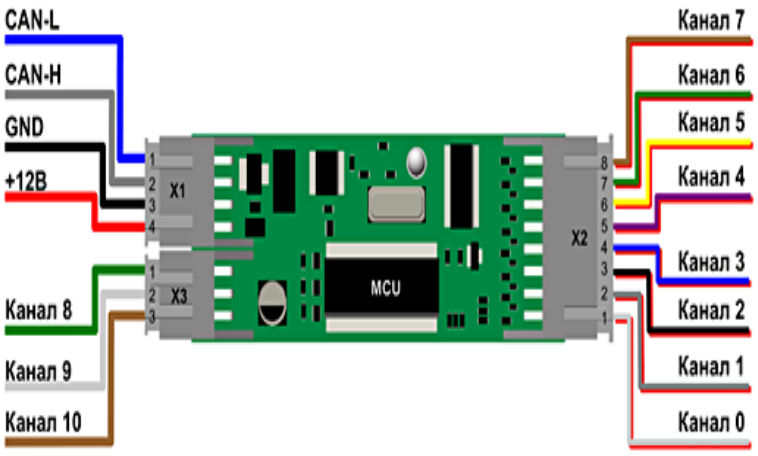

Рисунок 4.23 – Схема використання CANNY

Для підключення планується використання платформи контроллера CANNY, що дозволяє легко проводити підключення та програмувати компоненти мережі CAN.

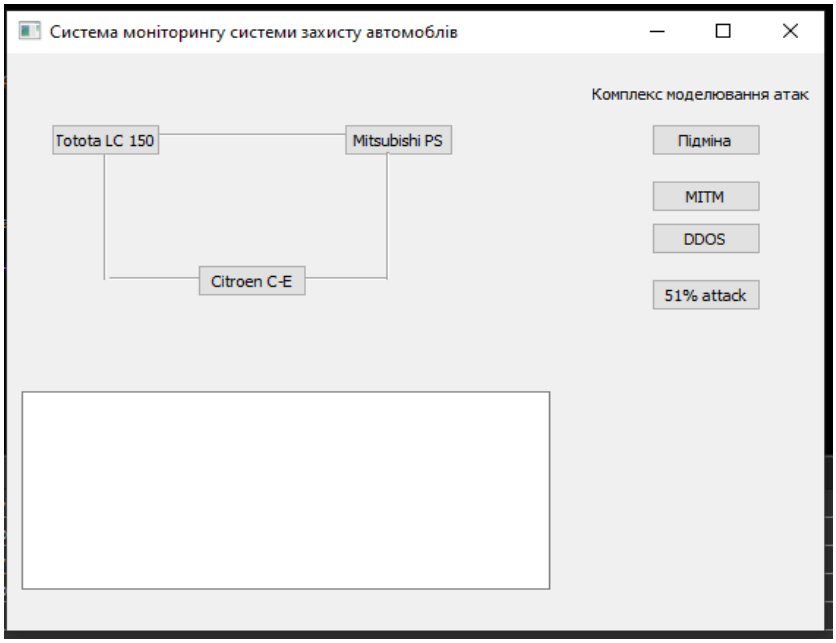

Рисунок 4.24 – Загальний вигляд програми

У програмі використовуються функції шифрування Калина, та гешування SHA-1 для забезпечення цілісності. На рисунку 4.24 можна побачити головне вікно розробленої системи.

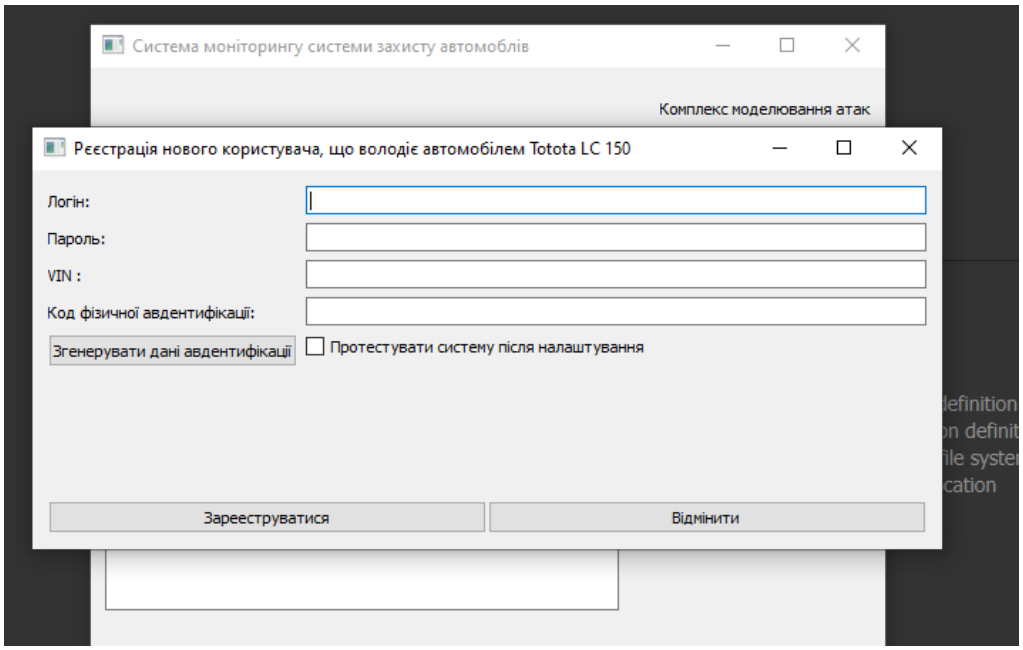

Рисунок 4.25 – Загальний вигляд програми

Для отримання можливості користування системою кожен користувач має пройти процес реєстрації(рисунок 4.25). При виборі в головному меню одного з доступних транспортних засобів користувачу буде показано вікно, в якому користувач має ввести бажаний логін, пароль, VIN(Vehicle Identification Number) – міжнародний ідентифікаційний номер автомобіля, та код фізичної автентифікації, що має бути виданий користувачу при купівлі автомобіля з встановленою системою.

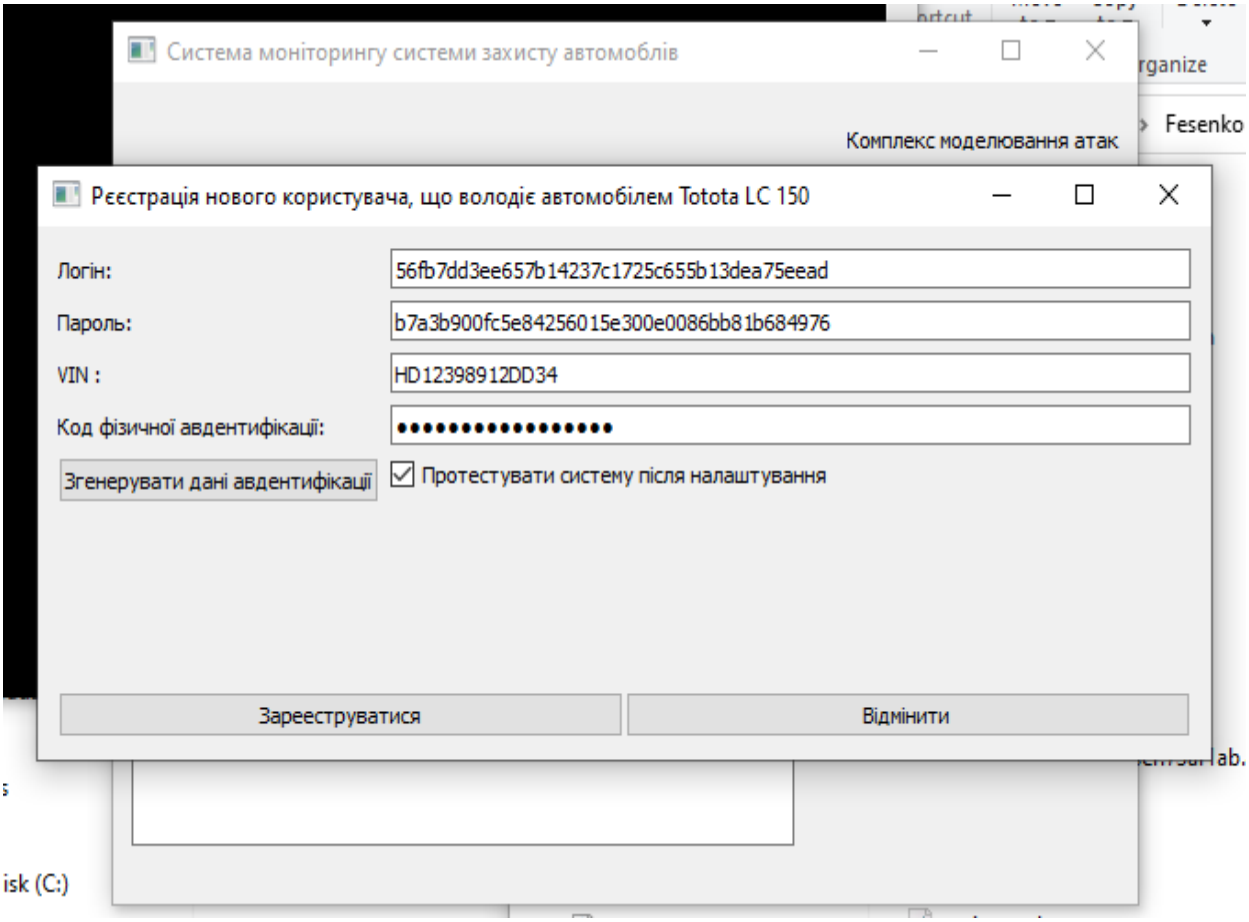

Рисунок 4.26 – Форма реєстрації нового користувача

Для спрощення було додано можливість автоматичної генерації пари логін-пароль з достатнім рівнем безпеки, після чого в базу даних сервера автентифікації буде додано обліковий запис, який буде зберігати дані в захищеному вигляді у формі геш-значень, що дозволять безпечно провести авдентифікацію користувача без використання критичних даних при передачі через мережу чи зберігання на сервері(рисунок 4.26).

Для можливості автентифікації через сервер автентифікації необхідно додати на сервер користувача, для цього необхідно ввести данні для автентифікації, зокрема логін, пароль.

Розглянемо також роботу «серверу» автентифікації використовується для генерації даних для ідентифікації блоків для доступу до криптоконтейнеру, що представляє собою блок імобілайзера.

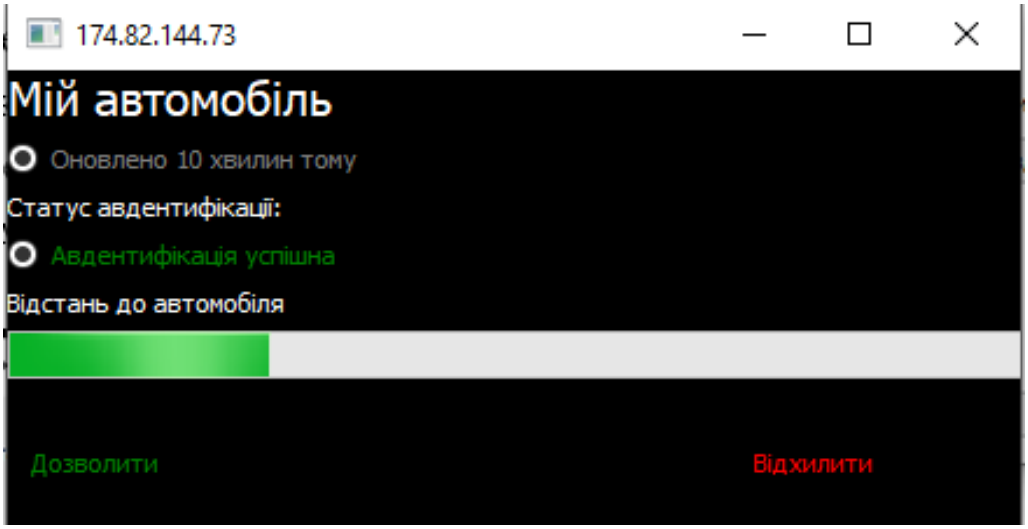

Рисунок 4.27 – Отриманий ключ від імобілайзера

Також необхідно ідентифікувати геопозицію GPS положення автомобіля та ідентифікатора сигналізації, отримане значення якого показано на рисунку 4.27.

При ініціалізації ключа з системою імобілайзеру, який в архітектурі має назву «keyless-go» починають свою роботу усі блоки автомобіля, в програмі ініціювати ключ можна за допомогою кнопки «Init keyless-go», після чого повинні вступити в дію індикатори «Keyless-go avalable», що дозволяє пристроям почати єтап автентифікації через блоки арбітражу. Для проведення автентифікації блоку загального призначення необхідно натиснути кнопку «Auth» в полі блоку «Multipurpose block», після чого на блок буде переданий псевдовипадкове сильне число у вигляді ідентифікатору для автентифікації на блок арбітражу, та визначено ідентифікатор самого блоку за статичними

методами маршрутизації протоколу CAN, який виступає в ролі керуючого протоколу для систем автомобіля. Отримавши дані про бажання автентифікації від блоку загального призначення та ідентифікувавши блок, арбітражному блоку необхідно надати докази свої повноважень щодо отримання доступу до ключів шифрування та надання доступу до них пов’язаному блоку загального призначення. Для цього арбітражний блок, що підключений до мережі інтернет відправляє запит на свою авдентифікацію використовуючи дані користувача, зокрема дані для входу, номер ідентетифікації автомобіля(VIN), мітку глобальної системи позиціювання GPS, на якого був зареєстрований даний транспортний засіб у базі сереверу автентифікації.

Арбітражний блок генерує пакет з даних використовуючи також мітку часу та псевдовипадкове значення для захисту від атак типу «повтор», від атаки підміна захищатися в даному випадку немає сенсу, оскільки дані, що передаються не представляюсть ніякого інтересу для зловмисника і їх модифікація лише призведе до руйнування цілісності даних і авдентифікація має бути проведена заново.

Для забезпечення захисту на сервері використовується можливість використання лише виділеної IP адреси, по яких фільтрується вхідний трафік.

Після отримання сервером автентифікації даних для ідентифікації та перевірки їх правильності для вказаного користувача генерується одноразовий PIN-код, час дії якого 15 секунд, зберігається на сервері у вигляді геш-значення, інформація про те, що код був згенерований передається на смартфон користувача, який за допомогою спеціального програмного забезпечення по захищеному каналу зв’язку отримує з сервера автентифікації PIN-код.

Після відправлення повідомлення про стан сгенерованості коду на мобільний телефон серевер відправляє зворотню відповідь до арбітражного блоку результатів автентифікації на сервері, з’єднання з мобільним телефоном та генерації PIN-коду у вигляді числового ідентифікатора.

На основі отриманих даних арбітражний блок визначає, чи необхідно повторити спробу автентифікації у випадку помилки, а у випадку успіху

пропонує ввести отриманий від програми на мобільному телефоні PIN-код за допомогою пристрою вводу інформації (розглядається можливість підключення системи попередження вторгнень як частини мультмедійної системи автомобіля для наочної демонстрації статусу безпеки та керування нею) у вікно для вводу даних. Після підтвердження вірності введення PIN-коду натисненням кнопки «ОК» арбітражний блок генерує пакет до серверу автентифікації, що має в собі дані про автомобіль(номер ідентифікації автомобіля(VIN), останні коррдинати положення GPS, ідентифікатор блоку імобілайзера який буде використовуватися для криптографічних операцій у майбутньому) та сам PIN-код у вигляді геш-значення. Після проходження автентифікації, сервер повертає відповідь про правильність операції, в випадку , згідно з якою арбітражний блок може отримати доступ до криптоконтейнеру у вигляді параметрів імобілайзера, та дозволяє також їх використовувати блоку загального призначення, який і ініціював з'єднання на початку роботи.

В режимі автоматизованої роботи авдентифікованих між собою блоків арбитражний блок знаходится завжди у режимі «прослуховування» ініціалізованих з'єднань, отримує дані від блоків загального призначення та передає іншому блоку. Блок загального призначення генерує пакет даних, що складається з ідентифікатору з'єднання, мітки часу генерації пакету, псевдовипадкового сильного числа, даних, що необхідно передати та геш-значення всього пакету.

Для забезпечення цілісності та секретності даних, що передаються використовується шифрування Калина-128, в якості ключа шифрування використовується строка, що складається з секретного ідентифікатора мітки ключа імобілайзера, який ініціалізується криптоконтейнером блоку імобілайзера, часу ініціалізації ключа в замку запалювання та часу останнього використання замка запалення(дані зберігаються в криптоконтейнері імобілайзеру) які конкатенуються в послідовність байтів розміру 128 біт. Пакет має статичний розмір з полями, що в суммі складають 128 біт. Це дозволяє

зашифрувати пакет без поділення на блоки, розмір блоку алгоритм Калина-128 це 128 біт, а розмір ключа також 128 біт.

Після проведення шифрування пакет передається арбітражному блоку, де той, генеруючи ключ за тим самим принципом, що і блок загального призначення і розшифровує отриманий пакет. Після отримання всіх даних блок верифікує правильність переданих даних за допомогою самостійної генерації геш-значення пакету та перевірки його з переданим значенням.

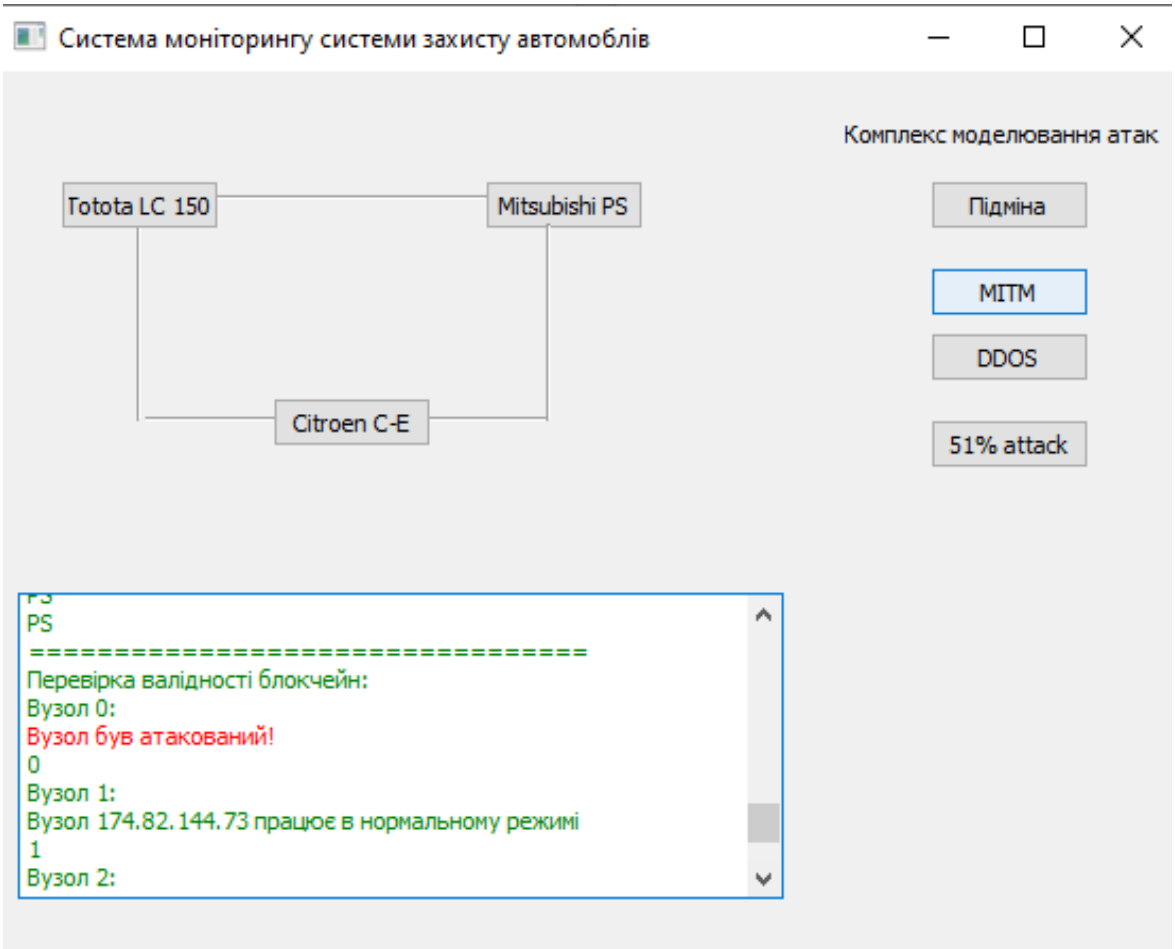

Рисунок 4.28 – Помилка при проведенні атаки типу «повтор»

У випадку успішно проведеної верифікації в логах арбітражного блоку буде показано повідомлення про те що пакет успішно валідований та може бути переданий до блоку призначення, що показано на рисунку 4.28.

Для перевірки можливості проведення атак на розроблений протокол імітуємо атаку типу «повтор», записавши раніше переданий пакет передамо його

на арбітражний блок повторно за допомогою натиснення на кнопку «Replay Attack», в наслідок чого арбітражний блок визначить, що псевдовипадковий ідентифікатор вже був нещодавно використаний та мітка часу просрочена, внаслідок чого буде показано повідомлення про те, що атака була викрита.

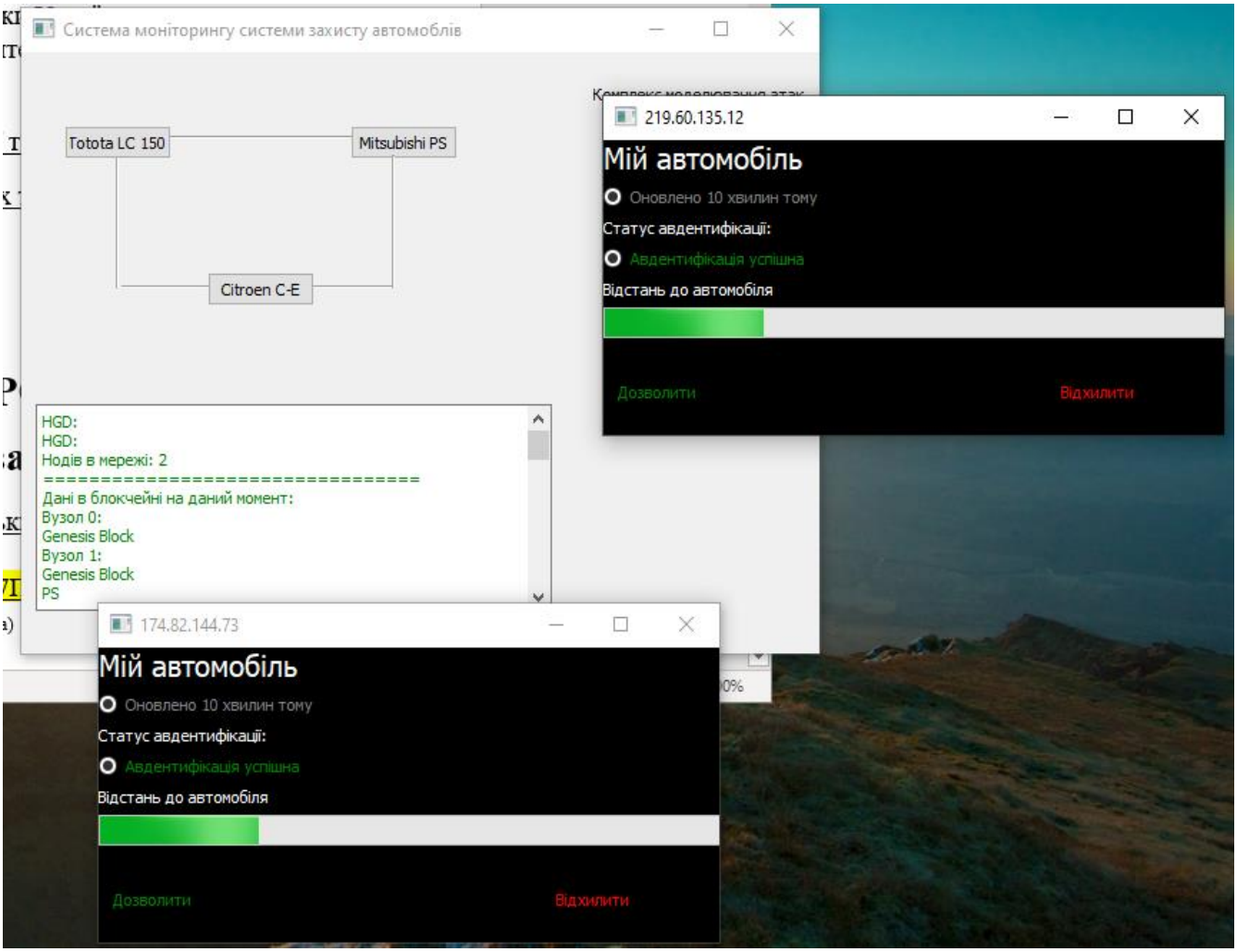

Рисунок 4.29 – Загальний вигляд робочого стану системи моніторингу

Інший вид атаки, що пропонується розглянути – це атака на підміну даних, коли атакуючий хоче підмінити один, або декілька байтів, її можна провести натиснувши кнопку «Data Attack», проведення якої було зафіксовано арбітражним блоком, оскільки геш-значення переданих даних не співпали переданим значенням.

На основі отриманих даних арбітражний блок визначає, чи необхідно повторити спробу авдентифікації у випадку помилки, а у випадку успіху

пропонує ввести отриманий від програми на мобільному телефоні PIN-код за допомогою пристрою вводу інформації (розглядається можливість підключення системи попередження вторгнень як частини мультмедійної системи автомобіля для наочної демонстрації статусу безпеки та керування нею) у вікно для вводу даних. Після підтвердження вірності введення PIN-коду натисненням кнопки «ОК» арбітражний блок генерує пакет до серверу авдентифікації, що має в собі дані про автомобіль(номер ідентифікації автомобіля(VIN), останні коррдинати положення GPS, ідентифікатор блоку імобілайзера який буде використовуватися для криптографічних операцій у майбутньому) та сам PIN-код у вигляді геш-значення. Після проходження авдентифікації, сервер повертає відповідь про правильність операції, в випадку , згідно з якою арбітражний блок може отримати доступ до криптоконтейнеру у вигляді параметрів імобілайзера, та дозволяє також їх використовувати блоку загального призначення, який і ініціював з’єднання на початку роботи.

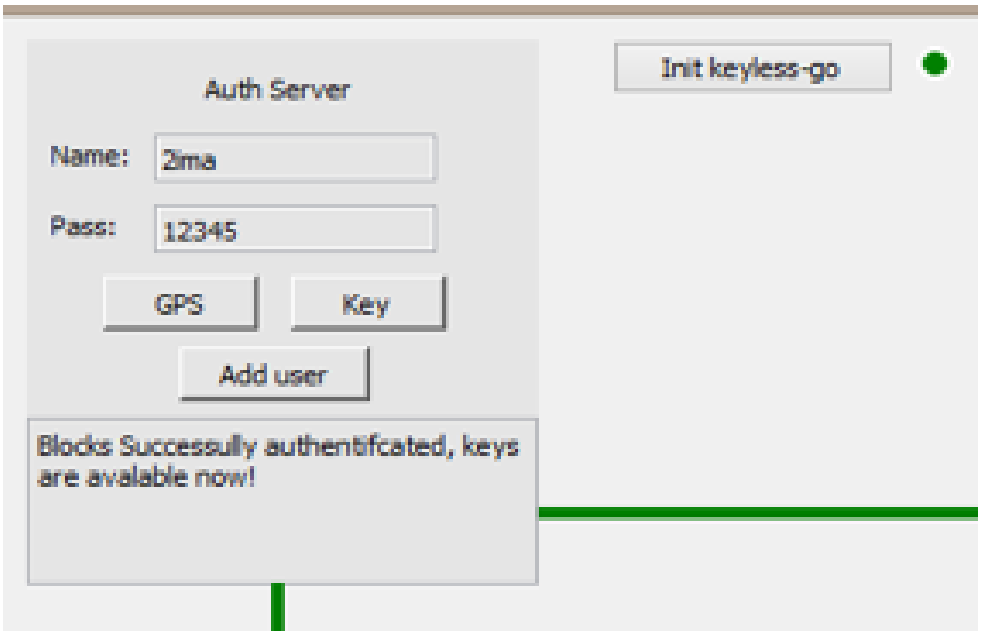

Рисунок 4.30 – Форма автентифікації на сервері

В режимі автоматизованої роботи авдентифікованих між собою блоків(на рисунку 4.30) арбитражний блок знаходится завжди у режимі «прослух-овування» ініціалі-зованих з’єднань, отримує дані від блоків загального призначення та передає іншому блоку. Блок загального призначення генерує

пакет даних, що складається з ідентифікатору з’єднання, мітки часу генерації пакету, псевдовипадкового сильного числа, даних, що необхідно передати та геш-значення всього пакету.

В якості ключа шифрування використовується строка, що складається з секретного ідентифікатора мітки ключа імобілайзера, який ініціалізується криптоконтейнером блоку імобілайзера, часу ініціалізації ключа в замку запалювання та часу останнього використання замка запалення(дані зберігаються в криптоконтейнері імобілайзеру) які конкатенуються в послідовність байтів розміру 128 біт.

Пакет має статичний розмір з полями, що в суммі складають 128 біт. Це дозволяє зашифрувати пакет без поділення на блоки, оскільки розмір блоку алгоритм Калина-128 це 128 біт, а розмір ключа також 128 біт.

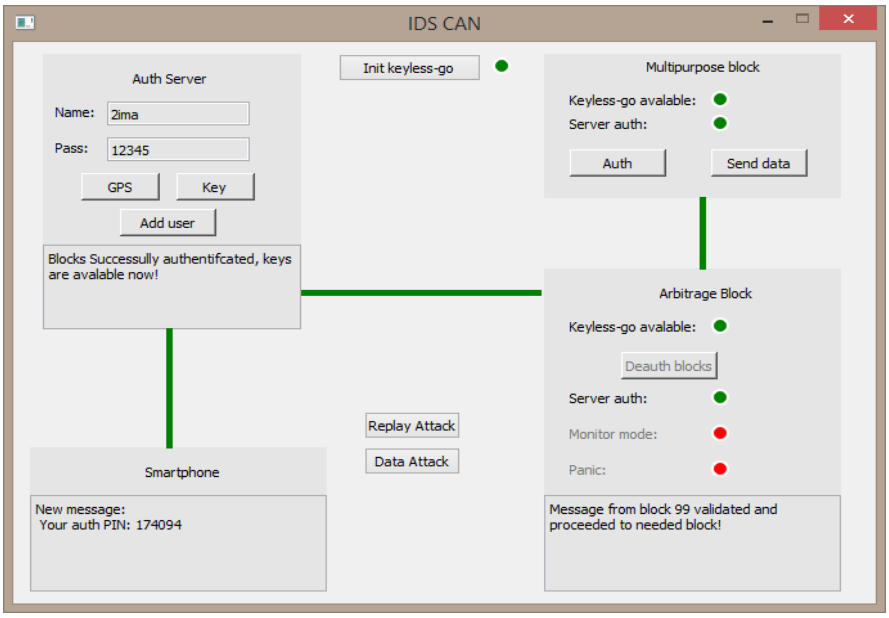

Рисунок 4.31 – Успішна передача даних між блоками

Після проведення шифрування пакет передається арбітражному блоку, де той, генеруючи ключ за тим самим принципом, що і блок загального призначення і розшифровує отриманий пакет.

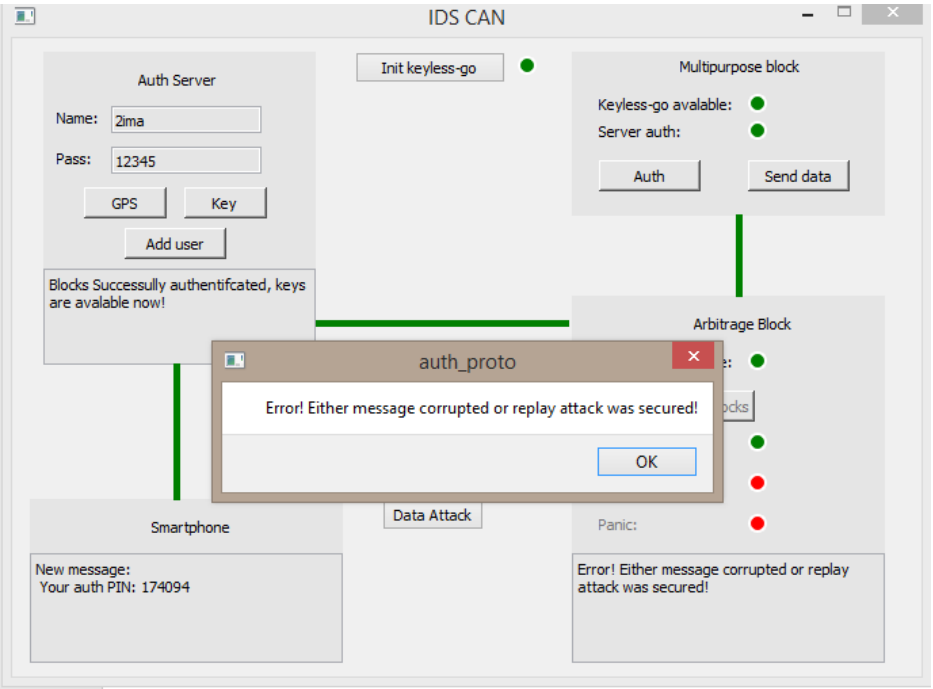

Рисунок 4.32 – Помилка при проведенні атаки типу «повтор»

Після отримання всіх даних блок верифікує правильність переданих даних за допомогою самостійної генерації геш-значення пакету та перевірки його з переданим значенням. У випадку успішно проведеної верифікації в логах арбітражного блоку буде показано повідомлення про те що пакет успішно валідований та може бути переданий до блоку призначення, що показано на рисунку 4.31.

Проведені тести атак на відключення вузлів автентифікації з мережі з використанням атаки повтор та підміна показали, що система може визначати проведення таких атак та не дозволяє неправомірний доступ до мережі автомобіля.

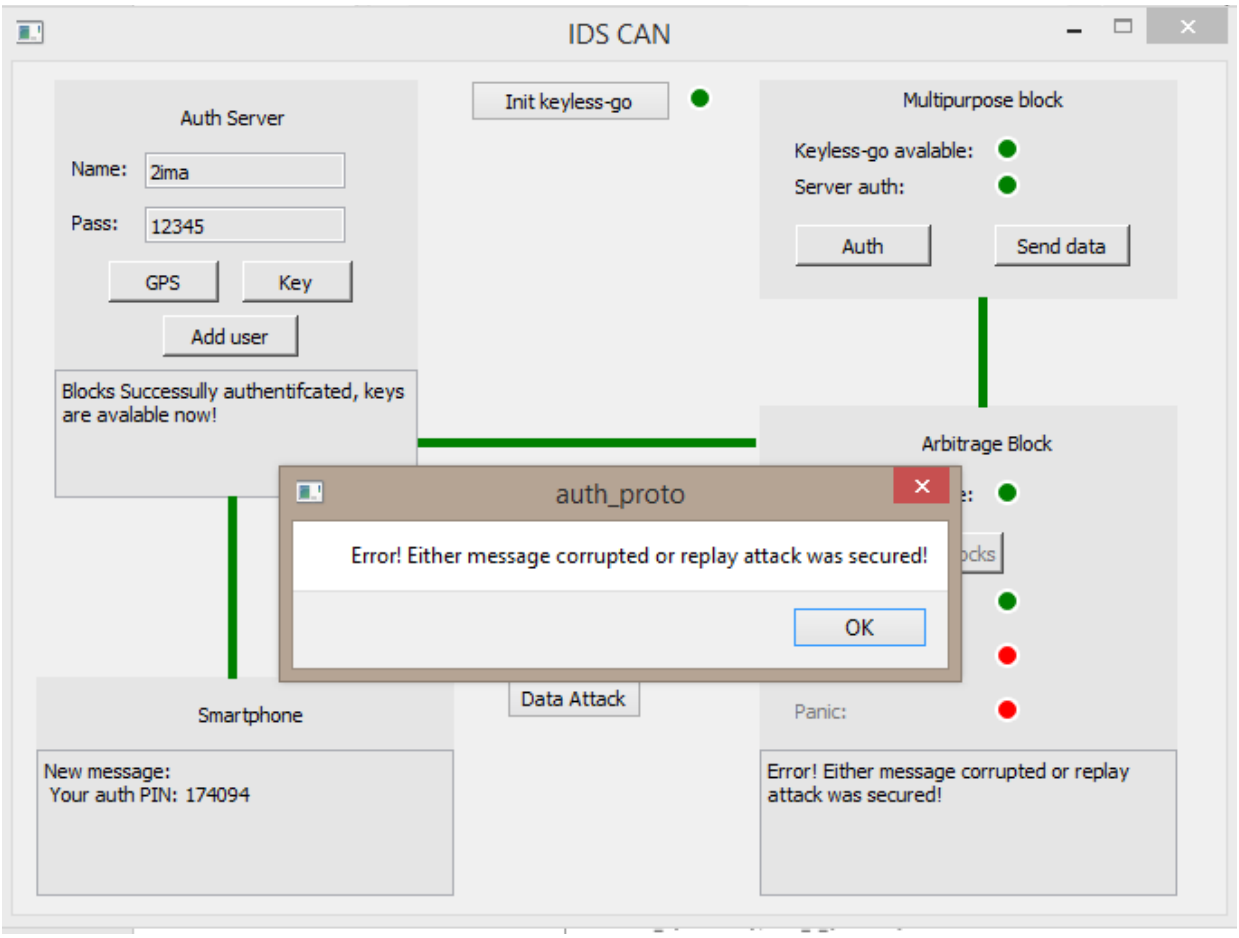

Рисунок 4.33 – Помилка при проведенні атаки типу «підміна»

Проведена атака DDOS, що повинна була перешкодити роботі мережі через виключення вузлу з мережі, але завдяки тому, що всі вузли були синхронізовані то атака була неефективною, атаки на отримання контролю над 51% блоків для нав’язування даних, що курсують в мережі, але навіть при невеликій кількості блоків, що тестувалися в цій атаці.

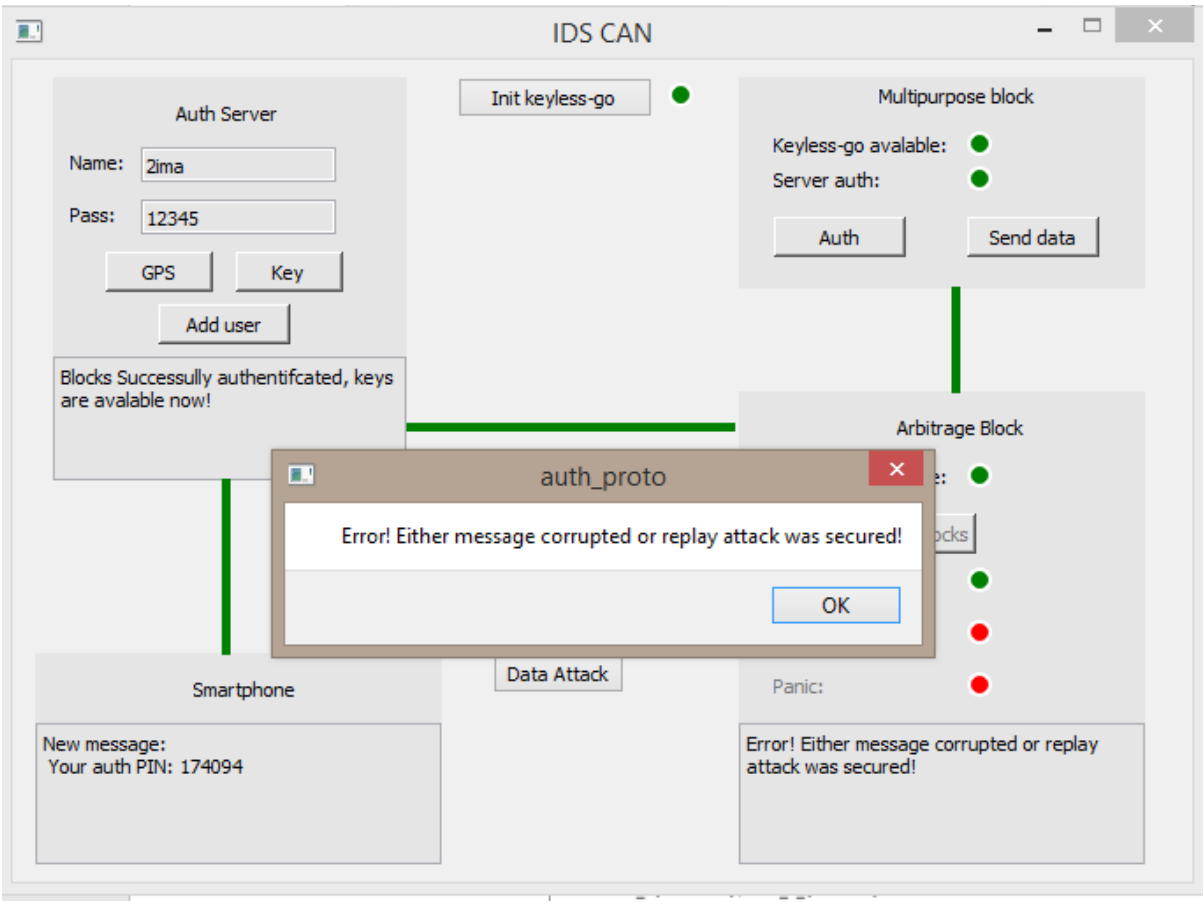

Рисунок 4.33 – Помилка при проведенні атаки типу «DDoS»

Атака стає ймовірною, коли зловмисником було отримано доступ до двох з трьох вузлів, та той отримав можливість робити нав'язування даних, але в той же час третій блок перестав потрібним чином працювати через те що не була пройдена перевірка валідності блоків, що не змогли синхронізуватися з скомпрометованими блоками (рисунок. 4.33).

Лістінг програми наведений у Додатку А. Програма була написана на мові C++ з використанням фреймворку QT.

4.5 Розгляд ключових частин програмної реалізації

Розглянемо детальніше деякі основні частини програмної реалізації. В програмній моделі використовується децентралізована мережа блокчейн, як було описано раніше блокчейн – послідовність блоків, що пов'язані між собою гешзначенням попереднього блоку, в програмній реалізації блок представлений у

вигляді класу, що має в собі поля, що зберігають дані про блок, їх склад, геш-значення та функції, що керують цими даними, розглянемо детальніше(повний код знаходиться в додатку А):

1) class CR_Block_BC : public QObject – в якості базового класу використовується об’єкт QObject для зв’язування викликів частини програми між собою

2) explicit CR_Block_BC(int index, QString data, UserData* currentCarInfo = nullptr, QString previouseHash="", QObject *parent = nullptr); – конструктор, який приймає на вході дані для створення нового блоку в мережі, де data – арбітражні дані, currentCarInfo – об’єкт, що має в собі інформацію про транспортний засіб.

3) CR_Block_BC(const CR_Block_BC &block) – конструктор копіювання, який на вході приймає все створений об’єкт

4) QString hash() const; – функція, що повертає геш блоку

5) void setHash(const QString &hash); – функція, що встановлює геш блоку(доступний тільки при ініціалізації)

6) int index() const; – функція, що повертає індекс блоку

7) void setIndex(int index);

8) qint64 timeStamp() const;

9) void setTimeStamp(const qint64 &timeStamp);

10) QString data() const; – функція, що повертає арбітражні дані блоку

11) void setData(const QString &data);

12) QString previouseHash() const; – функція, що повертає геш попереднього блоку мережі

13) void setPreviouseHash(const QString &previouseHash);

14) void mineBlock(int difficulty); – функція, що виконує майнінг блоку(вирахування значень)

15) QByteArray calculateHash();- функція, що вираховує геш-значення

16) int mIndex;

17) qint64 mTimeStamp; – часова марка інінціалізації ноди

18) QByteArray mData; - арбітражні дані

19) QByteArray mHash; - останнє геш-значення блоку

20) QByteArray mPreviouseHash; – кешоване геш-значення попереднього блоку мережі блокчейн

21) int mNonce;

22) QString userToAuthUID; – унікальний ідентифікатор користувача, який був авторизований для створення блоку

23) QByteArray userAuthData;

24) UserData* carInfo; – об’єкт, що зберігає дані про транспортний засіб для якого було створено блок

Головним відправником та отримувачем блоків(хабом, сервером) є нода мережі блокчейн, що зберігає з’єдання з іншими нодами в мережі, зберігає дані користувача, валідує інших користувачів:

1) explicit node_standalone(QWidget* my_ui, QObject *parent = nullptr); - конструктор для створення об’єкту вузла

2) void addBlock(CR_Block_BC*); – додає блок до цепи мережі

3) bool isValid(); – повертає статус валідності блоку

4) int difficulty() const; – повертає складність блоку

5) void setDifficulty(int difficulty); – встановлює складність блоку

6) void attack_block(int index);

7) CR_Block_BC* getLastBlock(){return mChain.at(mChain.length()-1);} – повертає останній блок мережі блокчейн, що був отриманий цією нодою

8) void setNodINdex(int index){nod_ind = index;} – дозволяє встановити для ноди унікальний ідентифікатор

9) int test() {return nod_ind;}

10) t_LChain getChain(){return mChain;} – повертає об’єкт мережі

11) void setChain(t_LChain chain){mChain = chain;}

12) void setNodeIP(const QString value_ip);- встановлює ай-пі адресу для ноди мережі

13) QString getNodeIP(){return node_ip;});- повертає ай-пі адресу для ноди мережі

14) void initChain(); – ініціалізує работу нода новим блоком

15) void initChain(CR_Block_BC* lastBlock); – ініціалізує работу нода на основі відомого останнього блоку.

16) t_LChain mChain;

17) int mDifficulty = 4; – складність блоку

18) QWidget* my_uiNode;

19) QString node_ip; – збережений ай-пі адреса

20) QList<node_standalone*> allConnectedNodes; – збереження всіх під’єднаних нодів об’єктів

21) int nod_ind;

22) protected:

23) void init();

24) t_LChain getCurrentChain(){return mChain;}

25) bool connectToNode(QString ip="localhost", QString name="test");

26) void newChainAdded(); – визивається коли якийсь інший вузол(нода) створила новий блок

27)

28) protected slots:

29) void addNewBlock(CR_Block_BC* block); – додає блок до кешу

30) void sendNewBlock(); – відправляє новий блок іншим учасникам

4.6 Висновки за четвертий розділ

За допомогою програмного моделювання роботи системи автентифікації було розглянено всі етапи роботи мережі системи, розглянуті механізми

генерації даних, їх синхронізації між собою, перевірки валідності даних та можливостей з їх модифікації, можливість впливу на дані автентифікації, що передаються, що дозволило зрозуміти захищеність каналу передачі даних, метою атаки було скомпрометувати дані, що зберігалися в попередніх блоках, але за рахунок перевірки блоків з використанням геш-значень попередніх блоків, що зберігаються в наступних блоках в блокчейні. Проведені тести атак на відключення вузлів автентифікації з мережі з використанням атаки DDOS, що повинні були перешкодити роботі мережі виключенням вузлу з мережі, атаки на отримання контролю над 51% блоків для нав'язування даних, що курсують в мережі, але навіть при невеликій кількості блоків, вона стала ймовірною, Таким чином, що розроблена програмна модель системи автентифікації автомобільної системи задовольняє вимогам.

ВИСНОВКИ

У ході роботи була розглянута побудова захищеної системи автентифікації для автомобільних систем, структура та будова комерційних пропозицій засобів безпеки для безпечної автентифікації власника автомобіля. Були визначені можливі загрози зі сторони зловмисників для системи автентифікації автомобіля, найнебезпечніші з цих загроз – модифікація даних, можливість проведення «реплей» атак, використання вразливих місць у коді мережі та зламування криптоалгоритмів. Була розроблена програмна реалізація тестової моделі систем безпеки з використанням функціоналу моделювання розглянутих в даній роботі видів атак.

Програмне моделювання роботи системи автентифікації дозволяє розглянути всі етапи роботи мережі системи, розглянути механізм генерації даних, їх синхронізації між собою, перевірки валідності даних та можливостей з їх модифікації. Було розглянуто можливість впливу на дані автентифікації, що передаються, за допомогою MITM атаки, що дозволило зрозуміти захищеність каналу передачі даних, метою атаки було скомпрометувати дані, що зберігалися в попередніх блоках, але за рахунок перевірки блоків з використанням геш-значень попередніх блоків, що зберігаються в наступних блоках в блокчейні. Були проведені тести атак на відключення вузлів автентифікації з мережі з використанням атаки DDOS, що повинна була перешкодити роботі мережі через виключення вузлу з мережі, але завдяки тому, що всі вузли були синхронізовані то атака була неефективною, атаки на отримання контролю над 51% блоків для нав’язування даних, що курсують в мережі, але навіть при невеликій кількості блоків, що тестувалися в цій атаці, вона стала ймовірною, коли зловмисником було отримано доступ до двох з трьох вузлів, та той отримав можливість робити нав’язування даних, але в той же час третій блок перестав потрібним чином працювати через те що не була пройдена перевірка валідності блоків, що не змогли синхронізуватися з скомпрометованими блоками. Таким чином, можна визначити, що

розроблена програмна модель системи автентифікації автомобільної системи задовольняє вимогам, висунутим під час розгляду захисту від атак.

Були представлені вимоги до рівня безпеки для безпечного використання систем автентифікації та пропозиції щодо досягнення відповідного рівня безпеки з використанням мережі блокчейн, розглянуті можливі загрози для системи безпеки при використанні в якості системи передачі даних мережу блокчейн.

За отриманими даними висунуті пропозиції щодо безпечного застосування технології блокчейн з огляду на розглянуті можливості атаки на реалізацію мережі блокчейн.

ПЕРЕЛІК ПОСИЛАНЬ

1. Lemke K. Embedded Security in Cars: Securing Current and Future Automotive IT Applications / Kerstin Lemke – K. : Springer, 273. – С. 50 – 250.

2. Brown. Vehicle Security Systems: Build Your Own Alarm and Protection Systems,: Newnes, 160, 1996. – С. 7 – 155.

3. Knight A., Hacking Connected Cars: Tactics, Techniques, and Procedures / Alissa Knight – K. : Information Systems, 2019 – С. 5 – 250.

4. CAN bus [Електронний ресурс]. – Режим доступу: www/ URL: https://en.wikipedia.org/wiki/CAN_bus – 22.04.2019 р.

5. Understanding CAN-Bus [Електронний ресурс]. – Режим доступу: www/URL:https://avscarsecurity.com/avs-car-security/choosing-a-caralarm/understanding-can-bus-systems – 26.04.2019 р.

6. Car Security 101 [Електронний ресурс]. – Режим доступу: www/ URL: https://www.lifewire.com/car-security-101-534872 – 01.05.2019 р.

7. NISTIR 8202. Блокчейн Technology Overview, 2017. – С. 7 – 20.

8. Don Tapscott, Alex Tapscott Блокчейн Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World / Don Tapscott, Alex Tapscott Блокчейн – K. : Information Systems, 2016 – С. 65 – 102.

9. Andreas M. Antonopoulos Mastering Bitcoin: Unlocking Digital Cryptocurrencies / Andreas M. Antonopoulos – K. : NGITS, 2014. – С. 10 – 150.

10. Блокчейн: атаки, безопасность и криптография [Електронний ресурс]. – Режим доступу: www/ URL: https://www.securitylab.ru/blog/personal/Informacionnaya_bezopasnost_v_detalyah/343072.php – 26.08.2018 р.

11. Прикладна криптологія. Теорія. Практика. Застосування : монографія / І.Д. Горбенко, Ю.І. Горбенко; Міністерство освіти і науки, молоді та спорту України, ХНУРЕ, ПАТ "ІІТ" – Харків, 2012.