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

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android”

Виконав здобувач вищої освіти
ІІ курсу, групи КІ-25М
ОПП «Комп’ютерна інженерія»
спеціальності F7 «Комп’ютерна інженерія»
_____ Графенюк В.О.
«____» _____ 2026 р.

Керівник проекту
доктор філософії (PhD)
_____ Усік П.С.
«____» _____ 2026 р.

Нормоконтроль проекту
_____ Коваленко А.С.
«____» _____ 2026 р.

Рецензент _____
_____

Підписано КЕП
Центральноукраїнський національний технічний університет
ГРАФЕНЮК ВАЛЕРІЯ ОЛЕКСАНДРІВНА
Сертифікат: 21E4B79083A82E7F04000000D0763C02D51AF607
Дійсний з 13.09.2026 18:21:49 по 14.09.2027 02:59:59

Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*

Кафедра *Кібербезпеки та програмного забезпечення*

Рівень вищої освіти *магістр*

Галузь знань _F_____*"Інформаційні технології"*_____

Спеціальність F7 *"Комп'ютерна інженерія"*

Освітньо-професійна (освітньо-наукова) програма _*"Комп'ютерна інженерія"*_

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 1 » вересня 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Графенюк Валерії Олександрівні
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи апаратно-програмного комплексу захисту каналів зв'язку стільникової мережі на системному рівні ОС Android*

2. Керівник роботи *Усік Павло Сергійович, доктор філософії (PhD)*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 41-13 від 17.08.2026 року

3. Строк подання студентом роботи до захисту *2.12.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою розробки є дослідження та програмна реалізація системи апаратно-програмного комплексу захисту каналів зв'язку стільникової мережі на системному рівні ОС Android*

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

1. Призначення та область використання.	*5. Впровадження системи в промислову експлуатацію*
2. Перегляд аналогічних існуючих систем.	*6. Наукова новизна.*
3. Опис і обґрунтування проектних рішень.	*7. Заходи з охорони праці та техніки безпеки.*
4. Етапи програмування системи.	*8. Висновки.*

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

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

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2026	16.11.2026

8. Дата видачі завдання « 1 » вересня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2026 р.	
3.	Розробка моделі компонента	20.10.2026 р.	
4.	Розробка структур даних	25.10.2026 р.	
5.	Розробка алгоритмів системи	30.10.2026 р.	
6.	Програмування алгоритмів	10.11.2026 р.	
7.	Розрахунки з охорони праці та техніки безпеки	15.11.2026 р.	
8.	Оформлення ПЗ	17.11.2026 р.	
9.	Попередній захист роботи	2.12.2026 р.	

Дата видачі завдання
« 1 » вересня 2026 р.

Підпис керівника

_____ _____
(прізвище та ініціали)

Завдання прийнято до виконання
« 1 » вересня 2026 р.

Підпис здобувача

_____ _____
(прізвище та ініціали)

АНОТАЦІЯ

Графенюк В.О. Дослідження та програмна реалізація системи апаратно-програмного комплексу захисту каналів зв'язку стільникової мережі на системному рівні ОС Android. F7 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи апаратно-програмного комплексу захисту каналів зв'язку стільникової мережі на системному рівні ОС Android.

Метою розробки є дослідження та програмна реалізація системи апаратно-програмного комплексу захисту каналів зв'язку стільникової мережі на системному рівні ОС Android.

Об'єктом дослідження є процес апаратно-програмного комплексу захисту каналів зв'язку стільникової мережі на системному рівні ОС Android.

Предметом дослідження є методи апаратно-програмного комплексу захисту каналів зв'язку стільникової мережі на системному рівні ОС Android.

Методи дослідження базуються на методах теорії побудови комп'ютерних мереж, методах теорії захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи апаратно-програмного комплексу захисту каналів зв'язку стільникової мережі на системному рівні ОС Android.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на мобільному пристрої з ОС Android.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, захист каналів зв'язку, стільникова мережа, системний рівень, ОС Android

ABSTRACT

Hrafeniuk V.O. Research and Software Implementation of a Hardware-Software System for Protecting Cellular Network Communication Channels at the Android OS System Level. F7 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

This master's-level final qualification thesis presents the development of software designed for a hardware-software system aimed at protecting cellular network communication channels at the Android OS system level.

The objective of this development is to research and implement a software-based solution for a hardware-software system designed to protect cellular network communication channels at the Android OS system level.

The object of the research is the process of protecting cellular network communication channels via a hardware-software system at the Android OS system level.

The subject of the research consists of the methods used within a hardware-software system to protect cellular network communication channels at the Android OS system level.

The research methodology draws upon the theory of computer network design, information security theory, mathematical statistics, and software development practices.

The outcome of the work is the software implementation of a hardware-software system for protecting cellular network communication channels at the Android OS system level.

During the development of the software model, an analysis of existing hardware and software tools was conducted. All components of the developed software are described in detail.

A user-friendly interface has been developed. Instructions for using the software are provided.

The software can be used on mobile devices running the Android operating system.

The software was developed in the Python environment.

Keywords: computer engineering, communication channel protection, cellular network, system level, Android OS

ВІДОМІСТЬ ПРОЕКТУ

		Текстові документи			
1	А4	Пояснювальна записка	81	–	
2	А4	Робоча програма	30	–	Додаток А
		Графічні документи			
3	А4	Наукова новизна	1	–	
4	А4	Структурна схема	1	–	
5	А4	Функціональна схема роботи системи	1	–	
6	А4	Діаграма взаємодії процесів	1	–	
7	А4	Блок-схема основної програми	1	–	
8	А4	Блок-схема роботи підпрограми	1	–	

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 9
ВСТУП..... 10
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 13
1.1 Призначення системи..... 13
1.2 Область застосування 14
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 17
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... 17
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування 22
2.3 Розгорнута постановка завдання 23
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 24
3.1 Опис функціонування системи 24
3.2 Розробка структурної схеми..... 29
3.3 Розробка функціональної схеми 41
3.4 Розробка діаграми процесів 43
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ 46
4.1 Розробка блок-схем та опис алгоритмів функціонування системи 46
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 57
6 НАУКОВА НОВИЗНА 62

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ 63
7.1 Вступ 63
7.2 Шкідливі і небезпечні фактори при роботі з комп’ютером 64
7.3 Аналіз санітарно-гігієнічних умов праці
на робочому місці користувача ПК 65
7.4 Розробка заходів з умов поліпшення охорони праці 68
7.5 Протипожежний захист 69
7.6 Розрахункова частина 70
8 ОСНОВНІ ВИСНОВКИ 72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 74

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

AP	–	точка доступу
DSSS	–	технологія Direct Sequence Spread Spectrum
EAP	–	протокол розширеної автентифікації Extensible Authentication Protocol
MIC	–	криптографічна контрольна сума
RADIUS	–	сервер доступу Remote Access Dial-in User Server
RC4	–	алгоритм шифрування
TKIP	–	протокол генерації генерація ключів WPA-2 шифрування даних Temporal Key Integrity Protocol
TLS	–	протокол захисту транспортного рівня Transport Layer Security
SSID	–	ідентифікатор мережі який передає точка доступу
WEP	–	Wired Equivalent Privacy – протокол безпеки
WPA-2	–	стандарт безпеки Wi-Fi Protected Access

ВСТУП

Актуальність теми. Оскільки пристрої Android стають все більш поширеними, ризики безпеки, пов'язані з ними, виходять за рамки програмних вразливостей і включають критичні слабкі місця апаратного забезпечення. Дана робота пропонує комплексний та систематичний огляд вразливостей, пов'язаних з апаратним забезпеченням, у системах Android, які можуть обійти навіть найскладніші програмні засоби захисту. У даній роботі збираємо та аналізуємо широкий спектр зареєстрованих вразливостей, представляючи нову систему категоризації для глибшого розуміння цих ризиків, класифікованих за ураженими апаратними компонентами, типом вразливості та потенційним впливом на безпеку системи. У даній роботі розглядаються ключові області, такі як недоліки управління пам'яттю, атаки по бічних каналах, небезпечний розподіл ресурсів систем-на-чіпі (SoC) та криптографічні вразливості. Крім того, в ній розглядаються можливі контрзаходи, включаючи апаратне шифрування, механізми безпечного завантаження та довірені середовища виконання (TEE), для зменшення ризиків, що виникають через ці апаратні загрози. Розглядаючи апаратні вразливості в рамках ширшої архітектури безпеки пристроїв Android, цей огляд підкреслює важливість апаратної безпеки для забезпечення цілісності та стійкості системи. Результати є цінним ресурсом як для дослідників, так і для фахівців з безпеки, пропонуючи розуміння розробки більш надійного захисту від нових апаратних загроз, з якими стикаються пристрої Android.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

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

– Огляд існуючих систем апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

– Дослідження системи апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

– Програмна реалізація системи апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

Об’єктом дослідження є процес апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

Предметом дослідження є методи апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

Методи дослідження базуються на методах теорії побудови комп’ютерних мереж, методах теорії захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

– Розроблено вітчизняний продукт апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVIII науково-технічній конференції здобувачів вищої освіти «Наука-виробництву», основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у роботі збірника праць молодих науковців ЦНТУ, випуск №17.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи апаратно-програмного комплексу захисту каналів зв'язку стільникової мережі на системному рівні ОС Android, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

У сучасному технологічному ландшафті безпека мобільних пристроїв є критично важливою, особливо враховуючи їхнє все частіше використання в таких секторах, як охорона здоров'я, фінанси та уряд, де конфіденційні дані та цілісність системи мають першочергове значення. Операційні системи (ОС), які виступають основою цих пристроїв, відіграють життєво важливу роль у забезпеченні їхньої функціональності та безпеки. У цій роботі розглядаються атаки на обладнання Android; тенденції поширення є лише мотивацією до розгляду Android з червня 2025 року по червень 2026 року, зосереджуючись на їхній частці ринку та наслідках цих тенденцій для ширшої цифрової екосистеми.

Згідно з даними Statcounter [1] Android залишається домінуючою операційною системою, з часткою ринку 40,17% у червні 2025 року, хоча вона дещо знизилася до 36,82% до червня 2026 року. Це домінування підкреслює важливість вивчення архітектури безпеки Android, особливо враховуючи його широке використання в мобільних пристроях та планшетах у численних галузях. Інші операційні системи, такі як Windows, iOS, OS X та Linux, демонструють різні тенденції, але не можуть конкурувати з повсюдною присутністю Android, особливо в мобільних секторах. Хоча вразливості програмного забезпечення були широко досліджені, останні події показують, що вразливості апаратного забезпечення стають дедалі актуальнішими. Це особливо актуально, оскільки мобільні пристрої зараз є невід'ємною частиною критично важливих секторів, таких як охорона здоров'я та фінанси, де будь-яке порушення безпеки може мати значні наслідки. Вразливості апаратного забезпечення принципово відрізняються від вразливостей програмного забезпечення тим, що вони спрямовані на фізичні компоненти пристрою, часто повністю обходячи програмний захист. Ці вразливості можуть поставити під загрозу всю модель безпеки пристрою, що призводить до більш серйозних та стійких порушень.

1.2 Область застосування

Нещодавні досягнення в галузі досліджень мобільної безпеки підкреслили критичну роль архітектур захисних систем, які інтегрують різні апаратні компоненти для значного покращення безпеки мобільних пристроїв, зокрема систем Android. Ці архітектури виходять за рамки програмних заходів безпеки, використовуючи апаратні засоби захисту, які усувають вразливості на більш фундаментальному рівні. У контексті мобільних пристроїв апаратні компоненти, такі як безпечне завантаження, TEE та апаратно-підтримувані механізми шифрування, відіграють ключову роль у захисті цілісності пристрою від широкого спектру складних атак.

Безпечне завантаження [2] є одним з фундаментальних рівнів цих архітектур. Воно гарантує, що під час процесу завантаження може запускатися лише перевірене та криптографічно підписане програмне забезпечення. Це запобігає зловмисникам втручатися в роботу завантажувача [3] або впроваджувати шкідливий код на критичному етапі роботи пристрою [4]. Перевіряючи цілісність системи під час запуску, безпечне завантаження зменшує ризик руткітів та інших постійних загроз, які можуть поставити під загрозу безпеку системи на апаратному рівні.

Крім того, TEE забезпечує захищену область процесора, яка ізолює конфіденційні операції від основної операційної системи. TEE особливо ефективні для захисту керування криптографічними ключами, автентифікації та інших критично важливих функцій безпеки. Забезпечуючи обробку цих операцій у безпечному, ізольованому середовищі, TEE зменшують поверхню атаки та запобігають несанкціонованому доступу до конфіденційних даних, навіть за наявності шкідливого програмного забезпечення або інших загроз в основній ОС. Дослідження, подібні до [5, 6], демонструють, як поєднання безпечного завантаження з TEE не тільки забезпечує безпечні процеси завантаження, але й ізолює критичні функції для підвищення загальної безпеки системи.

Ще одним ключовим аспектом цих захисних архітектур є апаратне шифрування [7], яке гарантує зберігання та обробку ключів шифрування в безпечному апаратному середовищі. Повне шифрування диска (FDE) [8] та шифрування на основі файлів (FBE) [9] – це поширені методи, що використовують апаратну безпеку для захисту даних користувача, що зберігаються. Ці механізми шифрування гарантують, що навіть якщо зловмисник отримає фізичний доступ до пристрою, він не зможе розшифрувати конфіденційні дані без належних апаратно-захищених ключів. Дослідження показали, що апаратне шифрування може значно зменшити ризик витоку даних, навіть у сценаріях, коли зловмисники намагаються обійти програмні засоби контролю безпеки.

Інтегруючи кілька апаратних компонентів у цілісну архітектуру безпеки, ці системи створюють багаторівневий захист, який є надійнішим, ніж традиційні програмні підходи [10]. Цей багаторівневий захист особливо ефективний для зменшення впливу складних векторів атак, таких як атаки по бічних каналах, експлойти пошкодження пам'яті та несанкціонований доступ до системних ресурсів. Наприклад, атаки по бічних каналах часто спрямовані на фізичні властивості пристрою, такі як його споживання енергії або електромагнітне випромінювання, для отримання конфіденційної інформації. Апаратне шифрування та технології TEE можуть маскувати ці фізичні сигнали, що ускладнює для зловмисників отримання корисної інформації за допомогою аналізу побічних каналів [11]. Аналогічно, вразливості пошкодження пам'яті, які часто використовуються в програмних атаках, важче використовувати, коли встановлені засоби захисту на апаратному рівні, такі як ізоляція пам'яті та безпечне керування пам'яттю.

Для порівняння, програмні вразливості зазвичай стосуються недоліків у програмах або самій операційній системі та, як правило, легше виправляються за допомогою оновлень або виправлень безпеки. Однак апаратні вразливості використовують слабкі місця у фізичних компонентах пристрою, таких як процесори, пам'ять або модулі зв'язку.

Ці атаки часто мають ширший вплив, оскільки вони можуть обійти кілька рівнів програмного захисту, роблячи традиційні заходи безпеки неефективними. Як результат, поверхня атаки, що піддається апаратним вразливостям, значно більша та важче захиститися порівняно з програмними вразливостями.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Android працює на понад 3 мільярдах активних пристроїв по всьому світу та займає приблизно 42 відсотки ринку смартфонів Україні. Це також мобільна платформа з найширшим спектром загроз – відкрита екосистема додатків, нестандартне завантаження, сотні виробників пристроїв, нерівномірні терміни оновлень та системні API VPN та DNS, які дозволяють справжнім програмам безпеки робити те, що вони не можуть робити на iOS. Останній пункт має більше значення, ніж більшість покупців усвідомлюють. Програма кібербезпеки на Android може дійсно захистити вас способами, які просто неможливі на iPhone.

Компроміс полягає в тому, що пошук такого застосунку в Play Store може призвести до плутанини. Половина перших результатів – це безкоштовні програми «VPN», які направляють ваш трафік через ненадійні проксі-мережі, а інша половина – антивірусні сканери старої школи, які збільшують розряд батареї, не вирішуючи загрози, з якими більшість користувачів Android стикаються у 2026 році. Цей посібник розповідає про те, що насправді має робити програма для кібербезпеки Android цього року, і чому CyberFence створено, щоб охопити все це в одній установці.

Якою буде реальна картина загроз для Android у 2026 році

Ігноруйте маркетинг антивірусів. Ось що насправді зараз ставить користувачів Android під загрозу або вразливість:

– Фішингові посилання в SMS, WhatsApp та електронній пошті – «смішинг», спрямований на сповіщення про доставку, банківські сповіщення та запити MFA, є вектором атаки номер один проти телефонів. Це працює, тому що

фальшива сторінка входу виглядає ідентично справжній на маленькому екрані.

– Перехоплення публічних та готельних Wi-Fi мереж – кожна мережа аеропорту, кав'ярні та готелю є ненадійною. Сучасний HTTPS забезпечує значний захист, але DNS-запити, метадані та будь-які інструменти, що використовують відкритий текст, все одно протікають.

– Відстеження інтернет-провайдерів та операторів – Verizon, AT&T та T-Mobile можуть бачити кожен домен, який ви відвідуєте через мобільні дані, а в Україні мають законне право продавати рекламодавцям сукупні дані перегляду.

– Шкідлива реклама та мережі відстеження – більшість «шкідливого програмного забезпечення», що розміщується на телефонах Android, тепер надходить через рекламні мережі та скрипти відстеження, вбудовані в легітимні програми та веб-сайти, а не безпосередньо через шкідливі програми.

– Підроблені VPN та утиліти в Play Store – Дослідники безпеки нещодавно виявили сотні безкоштовних VPN-додатків, які непомітно маршрутизували трафік користувачів через комерційні проксі-мережі або продавали дані про місцезнаходження.

– Витік облікових даних через витік даних – витік пароля з будь-якого сервісу, яким ви користуєтеся, протягом кількох годин намагається зламати ваші облікові записи Gmail, банківських рахунків та соціальних мереж.

Зверніть увагу, що класичний антивірус типу «встановіть цю програму, запустіть сканування» не вирішує жодного з шести найпоширеніших. Загрози, які фактично вражають користувачів Android у 2026 році, виникають у мережі, на рівні DNS та через фішинг, а не через шкідливі APK-файли, що зберігаються у ваших файлах.

CyberFence

CyberFence – це VPN-сервіс, що працює в Україні та допомагає захистити ваш веб-браузер від сторонніх осіб під час підключення. Web Shield допомагає блокувати відомі шкідливі веб-сайти. Оберіть захист, який відповідає вашим пристроям та потребам. Персональні плани VPN починаються від 7,99$ на місяць із щомісячною оплатою. Скасувати поновлення можна будь-коли. Breach

Monitor – це окрема платна функція або включена до Complete.

Чотири рівні, які має охоплювати справжній додаток для кібербезпеки для Android

1. Зашифрований VPN-тунель

Кожен пакет, що залишає ваш телефон, має бути зашифрований від пристрою до надійного сервера. Це захищає вас у готельному Wi-Fi, Wi-Fi в аеропортах, мережах кав'ярень та точках доступу громадського транспорту. Це також запобігає тому, щоб ваш оператор реєстрував та продавав вашу історію переглядів у мобільних даних.

Зокрема, на Android VPN API є глибоким та надійним – справжній системний «постійно увімкнений VPN» з функцією автоматичного відключення працює так, як і має бути. Шукайте шифрування AES-256-GCM, WireGuard або аналогічні сучасні протоколи, перевірену політику нульового логування та роботу в Україні (не лише на серверах України).

2. Фішинг на рівні DNS та блокування шкідливих програм

Цей рівень пропускає більшість VPN-додатків, і саме він рятує вас від фішингової помилки. Коли у ваших текстових повідомленнях надходить фальшиве банківське сповіщення, і ви натискаєте на посилання, телефон намагається знайти домен фішингової сторінки. Блокувальник на рівні DNS відмовляється розпізнавати відомі фішингові та шкідливі домени, тому фальшива сторінка входу взагалі ніколи не завантажується. Це працює в кожному браузері, кожному додатку для обміну повідомленнями та кожному поштовому клієнті на телефоні.

Веб-щит CyberFence саме такий. Він працює на рівні DNS через тунель VPN, тому захищає Chrome, Samsung Internet, Firefox, Gmail, Outlook, Messages, WhatsApp, вбудовані браузери Instagram та будь-яке інше місце, куди може потрапити посилання.

3. Блокування реклами та трекерів

Окрім конфіденційності, це ще й засіб контролю безпеки. Більшість шкідливих програм, що потрапляють на Android через шкідливі рекламні мережі

та скрипти відстеження, завантажені на легітимні сайти. Блокування таких програм на рівні DNS повністю усуває механізм доставки. Це також робить телефон швидшим та економить дані.

На Android блокування реклами та трекерів на рівні DNS працює на рівні всієї системи, коли проходить через VPN-тунель – знову ж таки, платформа iOS ускладнює цю роботу. Web Shield від CyberFence охоплює це разом із блокуванням фішингу.

4. Моніторинг порушень

Електронна адреса середньостатистичної людини фігурує у понад 10 відомих випадках витоку даних протягом її життя. Коли сервіс, яким ви користуєтеся, порушується, ваш пароль потрапляє до списків вилучення облікових даних, і зловмисники протягом кількох годин намагаються атакувати його в Gmail, банківських облікових записах та соціальних мережах. Сучасний додаток безпеки для Android повинен відстежувати вашу електронну пошту на наявність відомих випадків витоку даних і попереджати вас про необхідність швидкої зміни облікових даних.

CyberFence Breach Monitor вирішує цю проблему. Додайте свої основні адреси електронної пошти в застосунок, і ви отримаєте сповіщення того ж дня, якщо виявляться нові порушення.

Що ігнорувати

Найкращі списки «безпеки» в Play Store сповнені функцій, які були актуальними десять років тому та не вирішують реальні загрози 2026 року. Надайте їм низьку вагу:

– Сканери APK-файлів та «сканування на віруси» – Google Play Protect вже безперервно сканує встановлені програми. Другий сканер додає розрядження батареї, а не захист, якщо ви часто не завантажуєте сторонні програми.

– Пакети «Booster», «Cleaner» та «memory optimizer» – Android сам керує пам’яттю. Ці функції є маркетинговими.

– Безкоштовні VPN – хтось має платити за пропускну здатність. Якщо ви

не платите, ваш трафік майже напевно платить.

– Блокування програм та сховища фотографій у програмах безпеки – Android вже має PIN-коди на рівні програм та зашифроване сховище. Пакетні версії цих функцій рідко бувають такими ж потужними, як еквіваленти, вбудовані в ОС.

Як налаштувати CyberFence на Android за п'ять хвилин

1. **Встановіть CyberFence з Play Store** – знайдіть «CyberFence» або скористайтеся прямим посиланням із сайту CyberFence.

2. **Увійдіть та перегляньте доступні плани** . Підписки на веб-сайті платні; будь-яка пропозиція магазину додатків діє за умовами, зазначеними в магазині.

3. **Надайте дозвіл на VPN** – Android відобразить системне повідомлення із запитом на налаштування VPN-з'єднання. Це нормально.

4. **Увімкніть «Постійно увімкнений VPN» та «Блокування з'єднань без VPN»** – Налаштування Android → Мережа та Інтернет → VPN → CyberFence → натисніть на шестерню → увімкніть обидва. Це змусить телефон відмовитися надсилати будь-який незашифрований трафік, зокрема коли додаток закрито.

5. **Увімкніть Веб-щит у застосунку CyberFence** – це вмикає блокування фішингу, шкідливого програмного забезпечення та реклами/трекерів на рівні DNS.

6. **Додайте свої адреси електронної пошти до Breach Monitor** – ви отримаєте сповіщення того ж дня, якщо будь-яка з них з'явиться у нових випадках порушення безпеки.

Як порівнюються ціни

CyberFence коштує $7,99/міс. щомісяця або $88,21/рік щорічно – $7,35/міс., економія 8%. Це включає VPN, блокування DNS Web Shield (фішинг + шкідливе програмне забезпечення + реклама + трекери), моніторинг порушень та підтримку від команди з України. Одна підписка поширюється на всі пристрої, які ви використовуєте – Android, iOS, Windows, Mac – а не на окремий додаток чи доповнення для кожної платформи.

Для порівняння, багато конкурентів стягують окрему плату за VPN та рівень захисту від фішингу/трекера, а більшість «безкоштовних» програм безпеки непомітно монетизують ваші дані перегляду. Математика значною мірою свідчить на користь єдиної підписки, що керується в Україні, без логування, яка охоплює всі чотири рівні.

CyberFence – це VPN-сервіс, що працює в Україні та допомагає захистити ваш веб-браузер від сторонніх осіб під час підключення. Web Shield допомагає блокувати відомі шкідливі веб-сайти. Оберіть захист, який відповідає вашим пристроям та потребам. Персональні плани VPN починаються від 7,99$ на місяць із щомісячною оплатою. Скасувати поновлення можна будь-коли. Breach Monitor – це окрема платна функція або включена до Complete.

Найкращий додаток для кібербезпеки для Android у 2026 році – це не антивірусний сканер. Це інтегрований стек: зашифрований VPN із сучасними протоколами, блокування фішингу та шкідливого програмного забезпечення на рівні DNS, яке працює в кожному додатку, блокування реклами та трекерів у масштабах всієї системи, а також моніторинг порушень адрес електронної пошти, прив’язаних до ваших облікових записів. CyberFence поєднує всі чотири функції в одному встановленні, одній підписці та одній компанії – керований в Україні, без журналів, відповідно до рекомендацій HIPAA, NIST, CMMC та SEC. Встановіть його, увімкніть Always-on VPN, активуйте Web Shield, і область загроз Android зменшиться до того, що дійсно важливо.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Мова програмування Python є ефективним інструментом для швидкого прототипування та розробки службових утиліт у середовищі мобільної ОС Android. Завдяки високорівневим структурам даних, виразному синтаксису та об'єктно-орієнтованій парадигмі вона оптимально підходить для написання сценаріїв автоматизованого тестування, аналізу радіоінтерфейсу та моделювання

алгоритмів захисту каналів стільникового зв'язку.

Офіційні пакети та документація інтерпретатора доступні на веб-сайті, забезпечуючи доступ до широкого екосистемного середовища бібліотек. Завдяки своїй архітектурі інтерпретатор Python легко інтегрується з низькорівневими компонентами, написаними на C або C++, що дозволяє зв'язувати високорівневу логіку комплексу безпеки з системними бібліотеками Android та апаратними модулями мобільного пристрою.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

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

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу

успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

З огляду на зростання кількості атак на основі апаратного забезпечення та зростаючу залежність від мобільних пристроїв у середовищах з високими ставками, важливо зосередитися на цих типах вразливостей. Нещодавні дослідження почали визнавати важливість апаратної безпеки, але вони залишаються фрагментарними, а категоризація та розуміння цих вразливостей все ще перебувають на ранніх стадіях.

Внесок цієї роботі є трискладовим:

1. Комплексна каталогізація: Ми надаємо вичерпний перелік вразливостей, пов'язаних з апаратним забезпеченням, про які повідомлялося та публікувалися на сьогодні. Кожна вразливість детально описана, з акцентом на її вплив на пристрої Android. Цей каталог пропонує цінний ресурс для дослідників і практиків, щоб зрозуміти зростаючий спектр і серйозність вразливостей апаратного забезпечення на пристроях Android.

2. Власна категоризація: На основі ретельного аналізу ми пропонуємо нову систему категоризації цих вразливостей. Ця система розроблена для диференціації апаратних вразливостей від їхніх програмних аналогів, зосереджуючись на конкретному ураженому апаратному компоненті, характері вразливості та її потенційному впливі на загальну архітектуру безпеки. Такий структурований підхід сприяє систематичній пріоритезації апаратних вразливостей.

3. Контрзаходи та контекстуалізація: Для кожної виявленої вразливості апаратного забезпечення ми обговорюємо можливі контрзаходи та рішення. Крім того, ми надаємо детальний опис механізмів безпеки на пристроях Android, щоб контекстуалізувати вразливості в ширшому ландшафті безпеки.

Щоб ретельно усунути вразливості безпеки в пристроях Android, важливо спочатку зрозуміти архітектуру та механізми безпеки, що лежать в основі платформи. У цьому розділі висвітлено ключові компоненти ОС Android [12], з особливим акцентом на аспекти, пов'язані з апаратним забезпеченням, такі як безпечне завантаження, TEE [13] та апаратне шифрування, які відіграють вирішальну роль у захисті від атак на апаратному рівні.

Архітектура Android та функції безпеки

Android побудовано на ядрі Linux, що забезпечує необхідну основу для управління ресурсами, включаючи драйвери обладнання та забезпечення безпеки. Однак верхні рівні архітектури, такі як середовище виконання Android та фреймворк додатків, покладаються на основні механізми апаратної безпеки для захисту системи від атак низького рівня.

Одним з ключових елементів моделі апаратної безпеки Android є реалізація безпечного завантаження. Безпечне завантаження гарантує, що під час процесу завантаження завантажується лише перевірене та авторизоване програмне забезпечення, запобігаючи зловмисникам впроваджувати шкідливий код на апаратному рівні. Цей процес, керований за допомогою криптографічних підписів, захищає цілісність послідовності завантаження.

Крім того, сучасні пристрої Android використовують TEE, захищену область основного процесора, яка виконує ізольований код та захищені програми. Це гарантує, що конфіденційні операції, такі як автентифікація та керування криптографічними ключами, виконуються безпечно, окремо від основної операційної системи. Ця ізоляція спрямована на зниження ризику атак, які намагаються використати ОС Android на апаратному рівні. Однак використання TEE не позбавлене вразливостей. Як зазначалося в Muñoz et al. [14], у самих TEE було виявлено широкий спектр вразливостей, що наражає їх на різні форми атак. Ці вразливості підкреслюють, що, незважаючи на свої обіцянки безпеки, TEE все ще може бути скомпрометований, що ставить під сумнів його роль непроникного бар'єру в архітектурі безпеки.

Апаратне шифрування – ще одна важлива функція архітектури безпеки

Android. FDE та FBE використовують апаратні ключі для захисту даних користувача, що зберігаються, гарантуючи, що навіть якщо зловмисник отримає фізичний доступ до пристрою, дані залишаться зашифрованими та недоступними без відповідних ключів розшифрування.

Окрім апаратного та програмного захисту, захист на системному рівні відіграє вирішальну роль у підтримці конфіденційності зв'язку на пристроях Android. Цей захист передбачає розгортання протоколів безпеки, які забезпечують безпечну обробку даних на всіх рівнях операцій. Наприклад, у середовищах з високим рівнем безпеки заходи системного рівня на Android можуть включати конфігурації безпечного розгортання, які використовують спеціалізовані апаратні модулі, технологію безпечного анклаву та надійні сторонні програми для забезпечення цілісності та конфіденційності даних. Дослідження загальносистемного захисту, особливо для програм, що обробляють конфіденційні комунікації, є сферою, що приділяє все більше уваги, спрямованою на створення структур, що захищають комунікації від несанкціонованого доступу навіть у ворожих мережевих середовищах.

Огляд рівнів ОС Android

Хоча ОС Android складається з кількох рівнів, включаючи прикладний рівень, платформу прикладних програм та середовище виконання Android, дана робота зосереджена на аспектах системи, які найбільше стосуються апаратних вразливостей. Віртуальна машина Dalvik (DVM) [15] та її наступник, середовище виконання Android (ART) [16], забезпечують середовище, де програми працюють в ізольованих пісочницях, використовуючи апаратні функції для забезпечення безпеки. Крім того, проміжне програмне забезпечення та нативні бібліотеки Android тісно взаємодіють з апаратними компонентами, такими як процесори та датчики, які є потенційними цілями для апаратних атак.

З огляду на важливість апаратного забезпечення в системі безпеки Android, вразливості на цих рівнях, особливо ті, що обходять програмний захист, повинні бути вирішені комплексно. Тому детальне розуміння цих рівнів можна знайти в [17] для контекстуалізації апаратних вразливостей.

Ключові механізми безпеки в Android

Архітектура безпеки Android включає кілька механізмів, призначених для захисту як апаратного, так і програмного рівнів:

– Безпечне завантаження: гарантує, що під час процесу завантаження завантажується лише перевірене програмне забезпечення, запобігаючи несанкціонованим змінам прошивки.

– Надійне середовище виконання: ізолює конфіденційні операції, такі як керування криптографічними ключами та автентифікація, забезпечуючи безпечний простір виконання окремо від основної ОС.

– Апаратне шифрування (HBE): FDE та FBE покладаються на апаратні ключі для захисту даних користувача, що робить їх стійкими до атак, навіть якщо фізичний пристрій скомпрометовано.

Крім того, «пісочниця» програм гарантує, що програми працюють в ізольованих середовищах, зменшуючи потенційну можливість того, що одна скомпрометована програма вплине на всю систему. Механізм міжкомпонентної комунікації (ICC) [18] полегшує зв'язок між програмами, одночасно забезпечуючи суворий контроль доступу, що ще більше обмежує поверхню атаки.

Еволюція патчів безпеки в Android

Патчі безпеки відіграють життєво важливу роль у вирішенні як програмних, так і апаратних вразливостей на пристроях Android. Хоча програмні вразливості часто можна усунути за допомогою регулярних патчів, апаратні вразливості можуть вимагати більш масштабних рішень, таких як оновлення прошивки або модифікації апаратних компонентів. З моменту запровадження щомісячних бюлетенів безпеки у 2015 році Google підтримує суворий процес встановлення патчів, приділяючи більше уваги виправленню вразливостей, що виникають через апаратні недоліки.

Проект Treble [19], представлений в Android 8.0 Oreo, ще більше спростив процес оновлення, відокремивши ОС Android від апаратних реалізацій, що надаються конкретними постачальниками. Цей архітектурний зсув покращив

швидкість та узгодженість оновлень безпеки, особливо тих, що вирішують проблеми, пов'язані з апаратним забезпеченням. Постійні зусилля в цьому напрямку, поряд з удосконаленням дизайну апаратного забезпечення та механізмів безпеки, є важливими для зменшення майбутніх вразливостей.

Безпека пристроїв Android нерозривно пов'язана як із програмним, так і з апаратним захистом. Роблячи акцент на безпечному завантаженні, TEE та HBE, архітектура Android забезпечує надійний захист від атак на апаратному рівні. Однак, оскільки мобільні пристрої продовжують відігравати вирішальну роль у таких секторах, як охорона здоров'я та фінанси, усунення вразливостей апаратного забезпечення має залишатися пріоритетом, особливо враховуючи, що ці атаки можуть обійти навіть найсучасніші програмні засоби захисту.

3.2 Розробка структурної схеми

Мобільні пристрої, від недорогих до висококласних смартфонів, стикаються з різноманітними проблемами безпеки, що виникають через різницю в якості апаратного забезпечення, програмній підтримці та методах виробництва. Ці відмінності критично впливають на їхню схильність до вразливостей, пов'язаних з апаратним забезпеченням, що може суттєво поставити під загрозу безпеку пристроїв. Розуміння цих проблем є важливим для розробки надійних контрзаходів та забезпечення безпечної роботи мобільних пристроїв. У цьому розділі представлено поглиблений аналіз вразливостей апаратного забезпечення в пристроях низького та середнього класу, а також зростаючий вплив широкодоступних хакерських інструментів, які використовують ці слабкі місця.

Пристрої низького класу

Бюджетні смартфони особливо вразливі до ризиків безпеки, пов'язаних з апаратним забезпеченням, головним чином через заходи щодо зниження витрат, які ставлять під загрозу цілісність апаратного забезпечення. Кілька факторів сприяють підвищенню ризиків безпеки в цих пристроях:

Незахищені апаратні компоненти: Смартфони низького класу часто містять апаратні компоненти нижчої якості, як це спостерігалося в попередніх аналізах виробничих практик, орієнтованих на витрати [32]. Ці компоненти більш схильні до вразливостей на апаратному рівні, включаючи незахищені процеси завантаження та слабке управління пам'яттю, що є закономірністю, що спостерігається в оцінках якості пристроїв [33, 34, 35]. Як наслідок, такі вразливості, як переповнення буфера та доступ до пам'яті за межами дозволених значень, є поширеними, значною мірою через відсутність основних апаратних засобів захисту, що робить ці пристрої основними цілями для атак зловмисників.

Недостатня підтримка та оновлення програмного забезпечення: Низькобюджетні пристрої зазвичай отримують мінімальні оновлення програмного забезпечення та безпеки, часто обмежені двома роками або менше [36]. Ця відсутність підтримки робить їх вразливими як до існуючих, так і до нових апаратних експлойтів. Відсутність постійних оновлень прошивки для критично важливих апаратних компонентів робить їх вразливими як до відомих, так і до нових апаратних експлойтів. Крім того, застарілі завантажувачі та незахищена прошивка зберігаються на цих пристроях, що ще більше посилює їхню вразливість до атак.

Обмежене забезпечення якості та тестування безпеки: Через обмеження виробничих витрат, низькоякісні пристрої проходять обмежене тестування безпеки, що може сприяти виникненню недоліків конструкції, таких як слабкі реалізації безпечного завантаження [37]. Це призводить до постійних недоліків конструкції, таких як слабкі реалізації безпечного завантаження та неадекватні механізми контролю доступу до обладнання. Крім того, виробники часто нехтують реалізацією апаратних функцій безпеки, таких як модулі захисту від несанкціонованого доступу, що збільшує ризик фізичних атак.

Компроміси продуктивності, що впливають на безпеку: У конструкціях, чутливих до вартості, оптимізація продуктивності може переважати заходи безпеки, як це спостерігалося в нещодавніх дослідженнях компромісів апаратного забезпечення [38]. Це може призвести до ослаблення або видалення

ключових механізмів безпеки, таких як апаратне шифрування, що робить пристрій вразливим до атак, які використовують ці компроміси. Наприклад, знижений криптографічний захист або недостатня ізоляція критичних процесів є поширеними точками входу для зловмисників.

З огляду на ці фактори, пристрої низького класу становлять значний ризик для безпеки в мобільній екосистемі. Користувачі та розробники повинні визнавати ці обмеження та впроваджувати додаткові захисні заходи, такі як включення апаратних модулів безпеки (HSM) [39] або TEE, щоб пом'якшити ці вразливості, де це можливо.

Пристрої середнього класу

Смартфони середнього класу пропонують баланс між ціною та продуктивністю, але продовжують стикатися зі значними проблемами безпеки. Нижче ми виділимо основні фактори, що впливають на безпеку пристроїв середнього класу:

Покращене обладнання з постійними вразливостями: Хоча пристрої середнього класу використовують компоненти вищої якості, ніж моделі низького класу, певні вразливості, такі як у безпеці прошивки та бездротових інтерфейсах, залишаються поширеними [40, 41]. Вразливості прошивки, незахищені бездротові протоколи та слабкі місця в датчиках, таких як GPS та акселерометри, є поширеними в пристроях середнього класу, як задокументовано в попередніх оглядах класів пристроїв [42]. Незважаючи на більш суворі заходи контролю якості, пристрої середнього класу все ще можуть мати недоліки, що піддаються використанню, в таких областях, як безпечне завантаження та безпека прошивки.

Цикли оновлень програмного забезпечення та безпеки: Пристрої середнього класу зазвичай отримують оновлення протягом двох-трьох років, що коротше, ніж пристрої високого класу, що може зробити їх більш вразливими до пізно виявлених апаратних експлойтів [43]. Однак цей період підтримки коротший, ніж у пристроїв високого класу, що робить смартфони середнього класу більш вразливими до апаратних експлойтів, виявлених після завершення

циклу оновлень. Затримки з усуненням вразливостей прошивки ще більше подовжують цей період впливу, посилюючи пов'язані з цим ризики безпеки.

Тестування безпеки та контроль якості: Хоча пристрої середнього класу проходять більш ретельне тестування, ніж моделі низького класу, їм все ще може бракувати комплексних оцінок безпеки, характерних для моделей високого класу [41]. Ця прогалина може призвести до невирішених недоліків конструкції, таких як недостатній контроль доступу до критично важливих апаратних регістрів. Крім того, залишаються занепокоєння щодо вразливостей до атак побічних каналів та підробки датчиків, особливо в компонентах, які не включають розширені заходи безпеки, зазвичай присутні в преміальних пристроях. Ці проблеми підкреслюють необхідність постійного вдосконалення протоколів безпеки, що застосовуються до обладнання середнього класу.

Компроміси між продуктивністю та безпекою: Хоча пристрої середнього класу здатні підтримувати розширені заходи безпеки, вони можуть не мати таких функцій, як апаратне сховище ключів, через обмеження вартості [44]. Критичні компоненти, такі як апаратне сховище ключів або криптографічні модулі, стійкі до несанкціонованого доступу, часто доступні лише в преміальних моделях. Отже, смартфони середнього класу залишаються вразливими до складних атак, особливо тих, що спрямовані на протоколи бездротового зв'язку або використовують вразливості фізичного доступу.

Незважаючи на покращений рівень безпеки порівняно з моделями низького класу, смартфони середнього класу все ще потребують пильної уваги, особливо щодо своєчасного оновлення прошивки та захисту важливих апаратних інтерфейсів. Вкрай важливо використовувати всі доступні функції апаратної безпеки та забезпечувати послідовне оновлення прошивки для ефективного зменшення потенційних ризиків безпеки.

Пристрої високого класу

Мобільні пристрої високого класу, зазвичай флагманські моделі, оснащені складним апаратним забезпеченням та функціями безпеки порівняно з пристроями низького та середнього класу. Ці висококласні пристрої

використовують передові технології, такі як апаратне сховище ключів, модулі захисту від несанкціонованого доступу та високозахищені довірені середовища виконання (TEE). Загальні функції, такі як безпечне завантаження та повне шифрування на апаратному рівні, ще більше посилюють їхній захист від несанкціонованого доступу на апаратному рівні. Незважаючи на ці покращення, кілька факторів продовжують ставити під загрозу безпеку висококласних пристроїв.

Складні апаратні архітектури зі збільшеною поверхнею атаки: високопродуктивні пристрої інтегрують кілька складних апаратних шарів, які, хоча й підвищують продуктивність і безпеку, також вводять додаткові вектори атаки. Вразливості в довірених компонентах (наприклад, захищених анклавах) та вплив атак побічних каналів можуть підірвати сприйняту безпеку цих пристроїв, дозволяючи зловмисникам атакувати елементи, які раніше вважалися безпечними.

Тривалі цикли оновлення програмного та апаратного забезпечення: Високоякісні пристрої зазвичай мають довшими цикли підтримки програмного та мікропрограмного забезпечення порівняно з моделями нижчого рівня. Однак затримки з усуненням вразливостей обладнання, особливо в критичних компонентах, таких як процесори базової смуги частот або спеціалізовані мікросхеми безпеки, можуть залишати пристрої без захисту протягом тривалого часу. Такий тривалий захист особливо непокоїть у середовищах, де ці пристрої зберігають конфіденційні дані, наприклад, у корпоративних або урядових контекстах.

Прогалини в безпеці та забезпеченні якості в складних системах: Хоча високопродуктивні пристрої проходять ретельне тестування, підвищена складність їхніх апаратних та програмних екосистем може ускладнити виявлення та виправлення всіх недоліків безпеки. Навіть за умови вдосконалених методів забезпечення якості, вразливості в процесах безпечного завантаження, прошивці та апаратних інтерфейсах можуть зберігатися, що призводить до потенційних експлойтів, які підривають загальну архітектуру

безпеки.

Компроміси між продуктивністю та безпекою в розширених функціях: Для підтримки оптимальної продуктивності високопродуктивні пристрої можуть зіткнутися з компромісами у впровадженні певних заходів безпеки. Хоча вони часто підтримують апаратне шифрування та розширене керування ключами, підтримка високої продуктивності може обмежувати глибину цих захистів. Наприклад, такі функції, як постійне підключення або високошвидкісна обробка, можуть створювати незначні вразливості, які зловмисники можуть використовувати, особливо в середовищах з високим рівнем ризику.

Доступні інструменти для злому

Зростаюча доступність недорогих інструментів для злому ставить під загрозу безпеку мобільних пристроїв. Багато з цих інструментів можна легко отримати через загальнодоступні канали, що дозволяє навіть людям з обмеженими технічними знаннями використовувати вразливості, присутні в смартфонах як низького, так і середнього класу. Ця зростаюча доступність знижує бар'єр входу для зловмисників, збільшуючи ризик компрометації. Нижче ми наводимо аналіз деяких найпоширеніших інструментів для злому та їхнього потенційного впливу на мобільну безпеку, зосереджуючись на тому, як ці інструменти сприяють використанню певних вразливостей.

Rubber Ducky USB (https://shop.hak5.org/products/usb-rubber-ducky (дата звернення: 2 жовтня 2026 р.)), Bash Bunny (https://shop.hak5.org/products/bash-bunny (дата звернення: 2 жовтня 2026 р.)) та Flipper Zero (https://shop.flipperzero.one/ (дата звернення: 2 жовтня 2026 р.)): Ці пристрої використовують вразливості в незахищених USB-інтерфейсах, встановлюючи шкідливі корисні навантаження або маніпулюючи даними під час процесу завантаження. Пристрої низького класу із застарілою прошивкою та неадекватними заходами безпеки USB особливо вразливі до цих атак. Пристрої середнього класу також знаходяться під загрозою, якщо їм бракує суворого контролю доступу до USB-інтерфейсу або безпечних конфігурацій завантаження.

Proxmark3 (https://github.com/RfidResearchGroup/proxmark3/ (дата звернення: 2 жовтня 2026 р.)) та WiFi Pineapple (https://shop.hak5.org/products/wifi-pineapple (дата звернення: 2 жовтня 2026 р.)): Proxmark3 [45] спрямований на погано захищені NFC-чіпи, які часто недостатньо захищені як у пристроях низького, так і середнього класу. Аналогічно, WiFi Pineapple використовує незахищені бездротові протоколи, такі як слабо реалізований Wi-Fi або Bluetooth, для здійснення атак типу «людина посередині» або перехоплення конфіденційних даних. Пристрої без надійного бездротового шифрування або безпечного керування ключами особливо вразливі до цих інструментів.

OMG Cable: OMG Cable (https://docs.hak5.org/omg-cable (дата звернення: 2 жовтня 2026 р.)), шкідливо модифікований USB-кабель, може використовувати незахищені завантажувачі або отримувати доступ до функцій на рівні прошивки в мобільних пристроях. Пристрої низького класу, які часто використовують застарілу або незахищену прошивку, особливо піддаються ризику, оскільки їм бракує захисту, такого як перевірка цілісності прошивки або обмеження USB-портів.

Доступність цих хакерських інструментів підкреслює нагальну потребу в посиленні апаратної безпеки мобільних пристроїв. Як апаратні, так і програмні вразливості необхідно вирішувати єдиним чином, щоб запобігти їх використанню цими легкодоступними інструментами. Виробникам слід надати пріоритет інтеграції апаратних функцій безпеки, таких як безпечне завантаження та шифрування на апаратному рівні, на всіх рівнях пристроїв, щоб ефективно зменшити ці ризики.

Методологія

Це дослідження пропонує систематичний огляд для виявлення та аналізу вразливостей, пов'язаних з апаратним забезпеченням, у пристроях Android. Методологію було розроблено з акцентом на ретельність, суворість та дотримання встановлених рекомендацій щодо досліджень інформаційної безпеки. Для виявлення та відбору вразливостей було застосовано багатоетапний

підхід, що поєднує комплексний пошук як в академічних, так і в публічних базах даних, а також застосування суворих критеріїв включення та виключення, що завершується детальним аналізом кожної виявленої вразливості.

Стратегія пошуку була зосереджена на виявленні відповідних досліджень у провідних академічних репозиторіях, включаючи IEEE Xplore, ACM Digital Library, SpringerLink та ScienceDirect. Було використано точну комбінацію ключових слів, таких як «вразливості апаратного забезпечення Android», «безпека мобільних пристроїв», «апаратні експлойти» та «слабкі місця безпеки Android», щоб забезпечити пошук літератури, що стосується критичних аспектів безпеки апаратного забезпечення на пристроях Android. Одночасно було проведено консультації з публічними репозиторіями вразливостей, такими як Національна база даних вразливостей (NVD) [46] та Загальні вразливості та експозиції (CVE) [47], для збору задокументованих вразливостей, про які повідомлялося в реальних сценаріях. Ця стратегія використання двох джерел забезпечила надійний набір даних, який об'єднує як академічні висновки, так і практичні звіти про вразливості.

У нашій методології ми провели комплексне дослідження, використовуючи широкий спектр добре відомих джерел для аналізу вразливостей у системах Android, включаючи апаратне забезпечення, прошивку та компоненти TrustZone. Ця методологія була структурована навколо ключових джерел, таких як CWE, CVE [48], NVD [46], Exploit-DB [49], VULDB [50], CERT [51], IEEE, ACM, Springer та ScienceDirect, щоб забезпечити чітке розуміння вразливостей з різних точок зору.

База даних CWE (Common Weakness Enumeration) була використана для виявлення вразливостей, пов'язаних безпосередньо з TrustZone, апаратним забезпеченням Android та прошивкою Android. Наприклад, CWE надала детальні записи про вразливості TrustZone із загальною кількістю 3 конкретних висновків та 219 посилань на апаратне забезпечення Android, що робить її критично важливим джерелом для розуміння слабких місць на рівні системи на кристалі (SoC).

Крім того, записи CVE (загальні вразливості та ризики) дозволили нам заглибитися в вразливості Android за допомогою вичерпних кількісних даних, що склали, серед іншого, 8657 записів для Android та 18 для апаратного забезпечення Android. Наприклад, посилання CVE-2016-2454 заслуговує на особливу увагу, оскільки воно ілюструє, як виявлені проблеми пов'язані з апаратними компонентами. Ці набори даних допомагають зрозуміти глибину та різноманітність загроз, що впливають на різні компоненти Android.

NVD (Національна база даних вразливостей) надала загалом 8555 записів, пов'язаних з Android та його апаратними вразливостями, тоді як інші специфічні джерела, такі як Exploit-DB, також використовувалися для виявлення публічно відомих експлойтів. База даних VULDB надала 14 вразливостей TrustZone, підтверджуючи аналіз практичними прикладами експлойтів.

Окрім баз даних вразливостей, ми включили дані з шанованих академічних та технічних журналів. IEEE, ACM, Springer та ScienceDirect були використані як основні академічні джерела, що гарантувало, що наш аналіз враховує останні дослідницькі розробки. Зокрема, IEEE включив низку з 52 конференцій та 14 журналів, присвячених вразливостям апаратного забезпечення Android, TrustZone та вразливостям прошивки, що забезпечило ретельне академічне підґрунтя для нашого дослідження.

Щоб забезпечити релевантність та якість вибраних вразливостей, було встановлено набір критеріїв включення та виключення. Критерії включення вимагали, щоб дослідження або звіти були опубліковані в період з січня 2010 року по липень 2026 року, зокрема, стосувалися вразливостей, пов'язаних з апаратним забезпеченням, у пристроях Android, та містили детальні технічні описи. Звіти, що зосереджувалися виключно на вразливостях програмного забезпечення, яким бракувало технічної глибини або які дублювали раніше переглянуті записи, були виключені. Цей процес гарантував, що в аналіз було включено лише високоякісні та відповідні дані.

Для детального аналізу було обрано кілька CWE, які включають детальні описи та вектори атак, що забезпечують чітке розуміння типів вразливостей

безпеки та відповідних їм механізмів експлуатації. Прикладами є CWE-1189 [52], який описує неправильну ізоляцію спільних ресурсів у SoC, вразливу до спекулятивного виконання побічних каналів, та CWE-1240 [53], пов'язану з криптографічними примітивами, реалізованими ризикованими методами. Вектори атак, пов'язані з цими вразливостями, такі як недостатньо захищені облікові дані або використання слабких криптографічних протоколів, пропонують критичне розуміння того, як такі вразливості можуть бути використані зловмисниками.

Початковий етап дослідження включав комплексне вивчення всіх потенційно значущих вразливостей обладнання, перелічених у базі даних Common Weakness Enumeration (CWE). Аналіз розпочався з розгляду широкого спектру вразливостей, що охоплюють різні апаратні компоненти, функції безпеки та конфігурації системи. База даних CWE надала розширений перелік вразливостей, що дозволило команді скласти загальний огляд існуючих та нових проблем в апаратних системах Android.

Однак, щоб забезпечити цілеспрямований та ефективний аналіз, було важливо звузити коло вразливостей на основі релевантності, впливу та практичності в реальних середовищах Android. Це призвело до уточненого вибору вразливостей, зокрема тих, що безпосередньо впливають на апаратну безпеку компонентів систем-на-чіпі (SoC) Android, криптографічних реалізацій, управління пам'яттю та механізмів захисту.

З початкового вичерпного списку ми зосередилися саме на наступних вразливостях, які мають велике значення через їхній потенційний вплив на пристрої Android та їхній прямий зв'язок з функціональністю обладнання та засобами контролю безпеки.

Рішення зосередитися на цих конкретних вразливостях було зумовлене бажанням зробити вагомий внесок у розуміння апаратних вразливостей у середовищах Android, які становлять найбільші ризики для безпеки. Ці вразливості були обрані на основі високої ймовірності їх використання, серйозності їхнього потенційного впливу та практичних труднощів, які вони

створюють для захисту апаратного та прошивкового забезпечення Android. Крім того, під час фази аналізу було помічено, що багато сценаріїв атак на апаратне забезпечення, згаданих у літературі, зрештою призвели до цих конкретних CWE як критичних вразливостей. Наприклад, CWE-1189 [52] широко обговорювався в таких джерелах, як «Виклики та можливості для апаратно-допоміжних покращень безпеки в польових умовах» (IEEE) [54], «На шляху до реконфігурованого обладнання для виправлення помилок обладнання в польових умовах» [55] (ScienceDirect), «IP-тег: виявлення троянів на основі тегів у середовищі виконання 3PIP на платформах SoC» [56] (IEEE), «Ризики безпеки та стратегії їх пом'якшення: перспектива хмарних обчислень» [57] (IEEE), «Апаратне забезпечення Phi-1.5B: велика мовна модель кодує знання, специфічні для апаратної області» [58] (ACM) та «Не переймайся цим: до методів аналізу CWE на ранніх стадіях проектування обладнання» [59] (ACM). Ці посилання підкреслюють критичний характер CWE-1189 [52] та подібних вразливостей, підтверджуючи рішення зосередитися саме на цих CWE в нашому детальному аналізі.

Такий підхід забезпечив, що, хоча спочатку було розглянуто широкий спектр вразливостей апаратного забезпечення Android, остаточна увага залишалася зосередженою на тих слабких місцях, які мають критичні наслідки для цілісності системи, конфіденційності даних та надійності експлуатації. Зосереджуючись на цих вибраних CWE, це дослідження має на меті забезпечити глибший аналіз та розробити цільові стратегії пом'якшення наслідків, які можна було б застосувати для підвищення безпеки апаратного забезпечення Android.

Для подальшого кращого розуміння цих вразливостей було проведено структурований процес вилучення даних для кожної вибраної CWE. Він включав детальні описи, потенційні вектори атак, уражені апаратні компоненти та можливі засоби пом'якшення наслідків. Кінцева мета дослідження – запропонувати практичну інформацію, яку як дослідницька спільнота, так і фахівці з безпеки можуть використовувати для ефективного зменшення вразливостей обладнання.

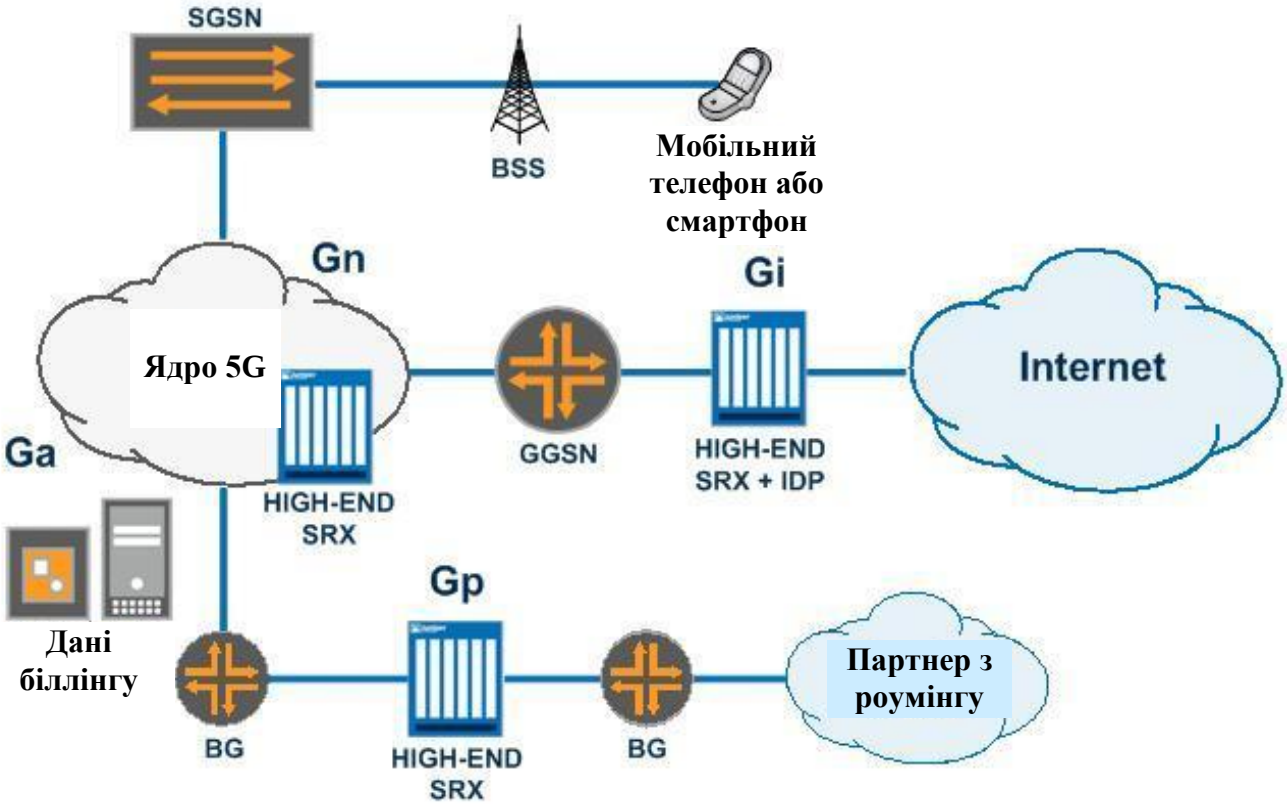

Рисунок 3.1 – Структурна схема системи

Після вилучення даних, вразливості були синтезовані, щоб забезпечити загальне уявлення про поточний ландшафт апаратних вразливостей на пристроях Android. Цей синтез включав детальний перелік вразливостей разом з оцінками впливу та рекомендованими контрзаходами. Інтегруючи вилучені дані, дослідження має на меті запропонувати чітке, практичне розуміння стану апаратної безпеки на пристроях Android, надаючи цінну інформацію як дослідницькій спільноті, так і фахівцям з безпеки.

Для забезпечення відтворюваності кожен етап методології, від початкового пошуку до остаточної категоризації вразливостей, був ретельно задокументований, включаючи детальний опис процесу відбору та обґрунтування конкретних вразливостей, проаналізованих з бази даних CWE. Такий структурований підхід гарантує, що результати цього огляду можуть бути надійно відтворені в майбутніх дослідженнях, тим самим забезпечуючи основу для постійних досліджень у цій критичній галузі безпеки пристроїв Android.

Завдяки систематичному та ретельному підходу це дослідження покращує розуміння апаратних вразливостей у пристроях Android. Результати є цінним ресурсом для дослідників та практиків у галузі мобільної безпеки, висвітлюючи критичні вразливості та пропонуючи потенційні шляхи їх усунення. Оскільки нові вразливості продовжують з'являтися, а система класифікації розвивається, ця робота закладає основу для майбутніх зусиль щодо підвищення безпеки пристроїв Android.

3.3 Розробка функціональної схеми

На рисунку 3.2 наведено функціональну схему розробленої системи (де AP позначає точку доступу / Access Point). Процес верифікації та встановлення захищеного з'єднання виконується за таким алгоритмом:

– Клієнтський пристрій ініціює відповідність між протоколом стільникового зв'язку та точкою доступу (AP).

– Точка доступу надсилає клієнту ідентифікаційний запит EAP (Extensible Authentication Protocol).

– Клієнт формує та передає на точку доступу свої облікові дані EAP, що містять ідентифікатор користувача та назву домену.

– Точка доступу пересилає отриманий запит на сервер RADIUS через неконтрольований мережевий порт.

– Після отримання облікових даних сервер RADIUS запитує сертифікат домену, що відповідає наданим реквізитам, та надсилає сертифікат сервера клієнту (через точку доступу).

– Клієнт верифікує отриманий серверний сертифікат та відправляє власний сертифікат користувача назад через точку доступу на сервер RADIUS.

– Сервер RADIUS виконує перевірку сертифіката за допомогою контролера домену Active Directory та Центру сертифікації (Certification Authority) задля гарантування відповідності облікових даних. У разі

невідповідності сервер надсилає точці доступу повідомлення про помилку автентифікації.

– При успішній верифікації сервер RADIUS надсилає точці доступу підтвердження разом із генераційним ключем стандарту WPA2 для поточного сеансу.

– Точка доступу ретранслює ключову інформацію WPA2 клієнтському пристрою. Цей ключ використовується впродовж усього сеансу зв'язку між клієнтом та AP.

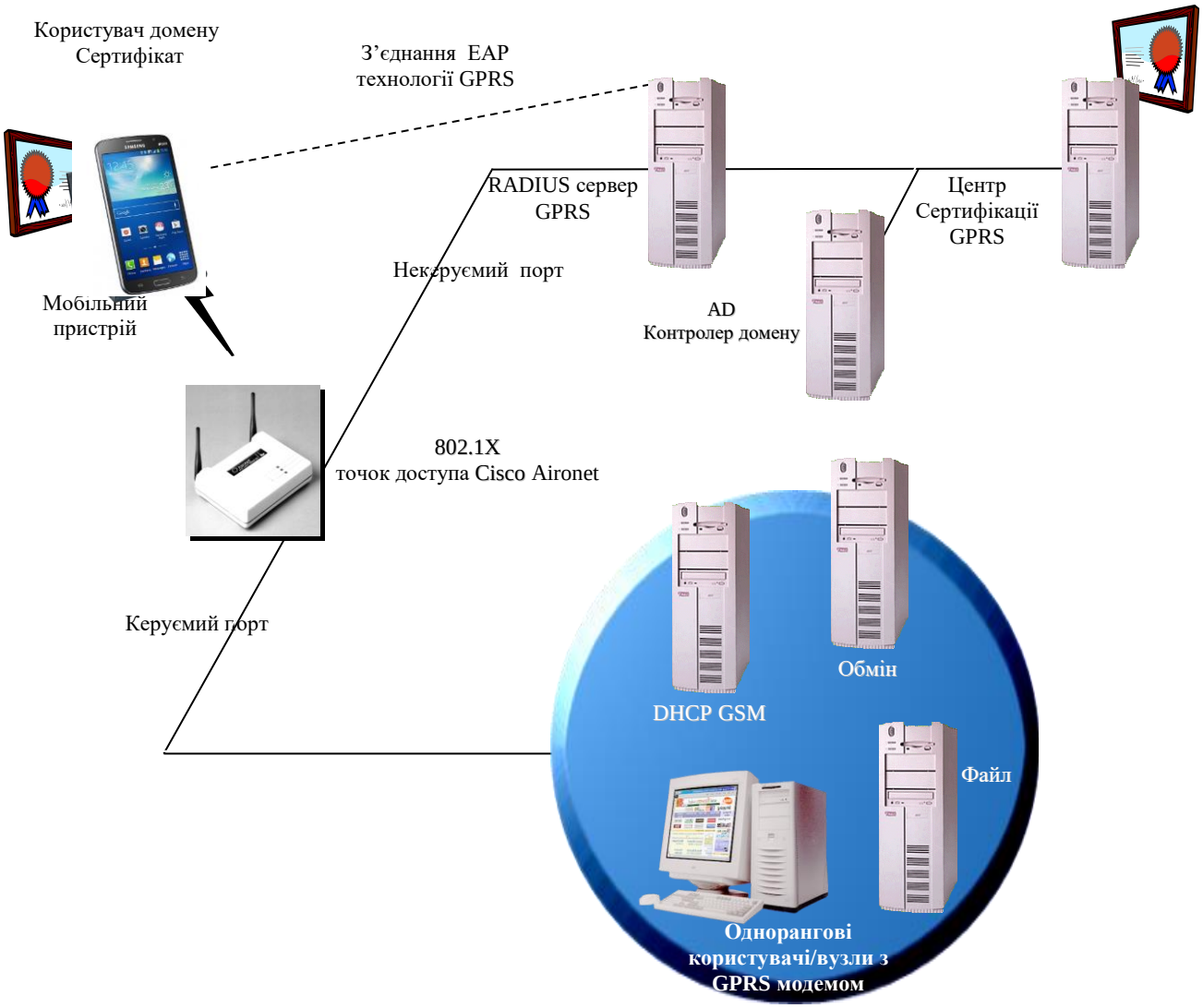

Рисунок 3.2 – Функціональна схема системи

– Точка доступу відкриває контрольований порт для надання клієнту доступу до мережних ресурсів.

– Клієнт застосовує ключ WPA2 для шифрування захищеного каналу зв'язку з AP та ініціює протокол DHCP для отримання коректної IP-адреси.

– Після успішного призначення IP-адреси DHCP-сервером здійснюється стандартна процедура входу в домен, після чого користувач отримує повноцінний доступ до захищеної мережі.

Протокол DHCP у системі розподілу мережних параметрів

DHCP (Dynamic Host Configuration Protocol – протокол динамічної конфігурації вузла) – це мережний протокол прикладного рівня, який забезпечує автоматичне налаштування IP-адрес та суміжних параметрів, необхідних для функціонування вузлів у мережах TCP/IP.

Протокол реалізує архітектуру «клієнт-сервер». На етапі ініціалізації мережного інтерфейсу клієнтський пристрій звертається до DHCP-сервера для автоматичного отримання конфігураційних даних. Мережний адміністратор визначає діапазони адрес (пули), які динамічно розподіляються між підключеними пристроями, що дозволяє мінімізувати ймовірність ручних помилок конфігурації та ефективно керувати адресним простором у корпоративних мережах.

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

3.4 Розробка діаграми процесів

На рисунку 3.3 наведено діаграму процесів розробленої системи. Після ініціалізації програмного забезпечення керування передається на головний модуль комплексу, у межах якого послідовно виконуються такі етапи та процедури:

1. Головне вікно ПЗ – ініціалізація та відображення основного інтерфейсу користувача і керуючих елементів системи.

2. Автентифікація користувача – перевірка облікових даних для доступу до захищеного середовища мобільного пристрою.

3. Створення базового ключа (стандарт 802.1X) – генерація початкового головного ключа (Master Key) на основі протоколу мережевої автентифікації 802.1X.

4. Передача ключа точці доступу (протокол TKIP) – трансляція ключової інформації до точки доступу з використанням протоколу тимчасового цілісного ключа (Temporal Key Integrity Protocol).

5. Передача ключа користувачу (протокол TKIP) – доставка сеансових параметрів на клієнтський пристрій через захищений канал TKIP.

6. Побудова ієрархії ключів (протокол TKIP) – генерація похідних сеансових ключів шифрування та контролю цілісності на основі узгодженого протоколу.

7. Генерація ключів шифрування – створення кінцевих криптографічних ключів для захисту переданих кадрів у стільниковій мережі.

8. Шифрування пакетів даних – апаратно-програмне перетворення вихідного інформаційного потоку з метою запобігання несанкціонованому перехопленню.

9. Контроль цілісності (алгоритм MIC) – формування коду цілісності повідомлення (Message Integrity Code, MIC) для захисту пакетів від модифікації.

10. Передача даних – трансляція зашифрованих і захищених пакетів через радіоінтерфейс стільникової мережі.

11. Дешифрування пакетів даних – зворотне криптографічне перетворення та розшифрування прийнятої інформації на приймальній стороні.

12. Перевірка цілісності (алгоритм MIC) – верифікація контрольної суми пакету для підтвердження відсутності спотворень у каналі зв'язку.

13. Завершення сеансу зв’язку – закриття активних захищених з'єднань, очищення оперативної пам'яті від чутливих ключів та деініціалізація протоколів безпеки.

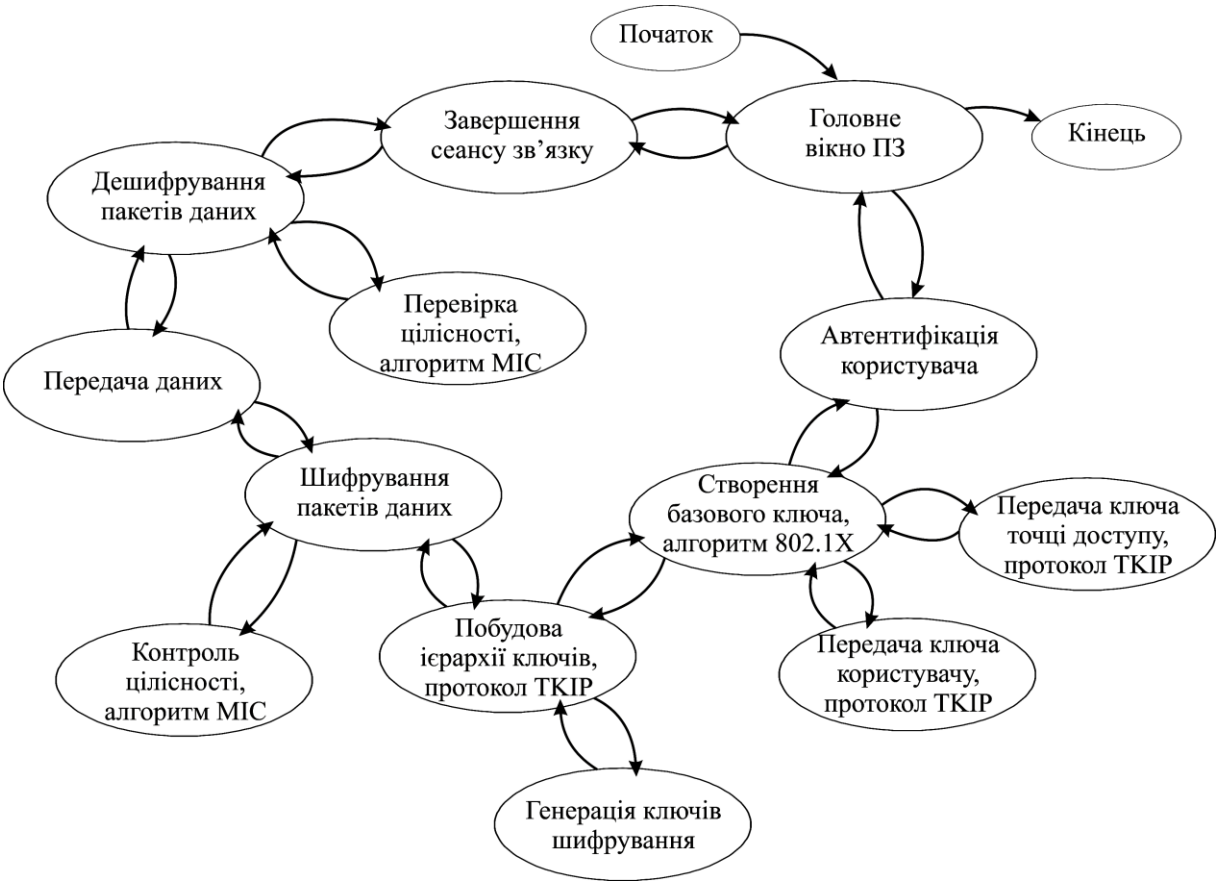

Рисунок 3.3 – Діаграма взаємодії процесів

У процесі проектування використовується модель структурного аналізу та проєктування, яка базується на графічному представленні потоків даних в інформаційній системі (Data Flow Diagram, DFD). Діаграма взаємодії процесів застосовується для візуалізації маршрутів обробки та передачі інформаційних масивів між компонентами комплексу. Згідно з інженерною методологією, проєктування розпочинається з розробки контекстної діаграми (діаграми нульового рівня), яка визначає глобальні межі системи та її інтерфейси із зовнішнім середовищем. Надалі ця модель підлягає ітераційній декомпозиції шляхом деталізації окремих процесів, протоколів захисту та потоків даних з метою повного відображення архітектури розроблюваного програмно-апаратного комплексу.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розроблення апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android передбачає створення механізмів контролю мережевого обміну, захищеного передавання інформації та перевірки стану мобільного пристрою.

Алгоритмічне забезпечення комплексу повинно враховувати особливості архітектури Android, розмежування доступу до системних ресурсів і можливості використання апаратних засобів криптографічного захисту.

Основним завданням є організація контрольованого проходження IP-трафіку через захищений канал, виявлення порушень установленої політики безпеки та запобігання несанкціонованому передаванню даних. Для реалізації цих функцій пропонується використати системний VPN-інтерфейс, механізми апаратно захищеного зберігання ключів і програмний модуль керування мережевими з'єднаннями.

Загальна послідовність функціонування комплексу включає ініціалізацію системних компонентів, перевірку доступності криптографічних засобів, завантаження політики безпеки, встановлення захищеного з'єднання та оброблення мережевих пакетів.

Основну блок-схему функціонування програмної системи передбачено на рисунку 4.1.

Архітектура розділяє підсистему керування, модуль контролю трафіку, криптографічний компонент і засоби взаємодії зі стільниковою мережею.

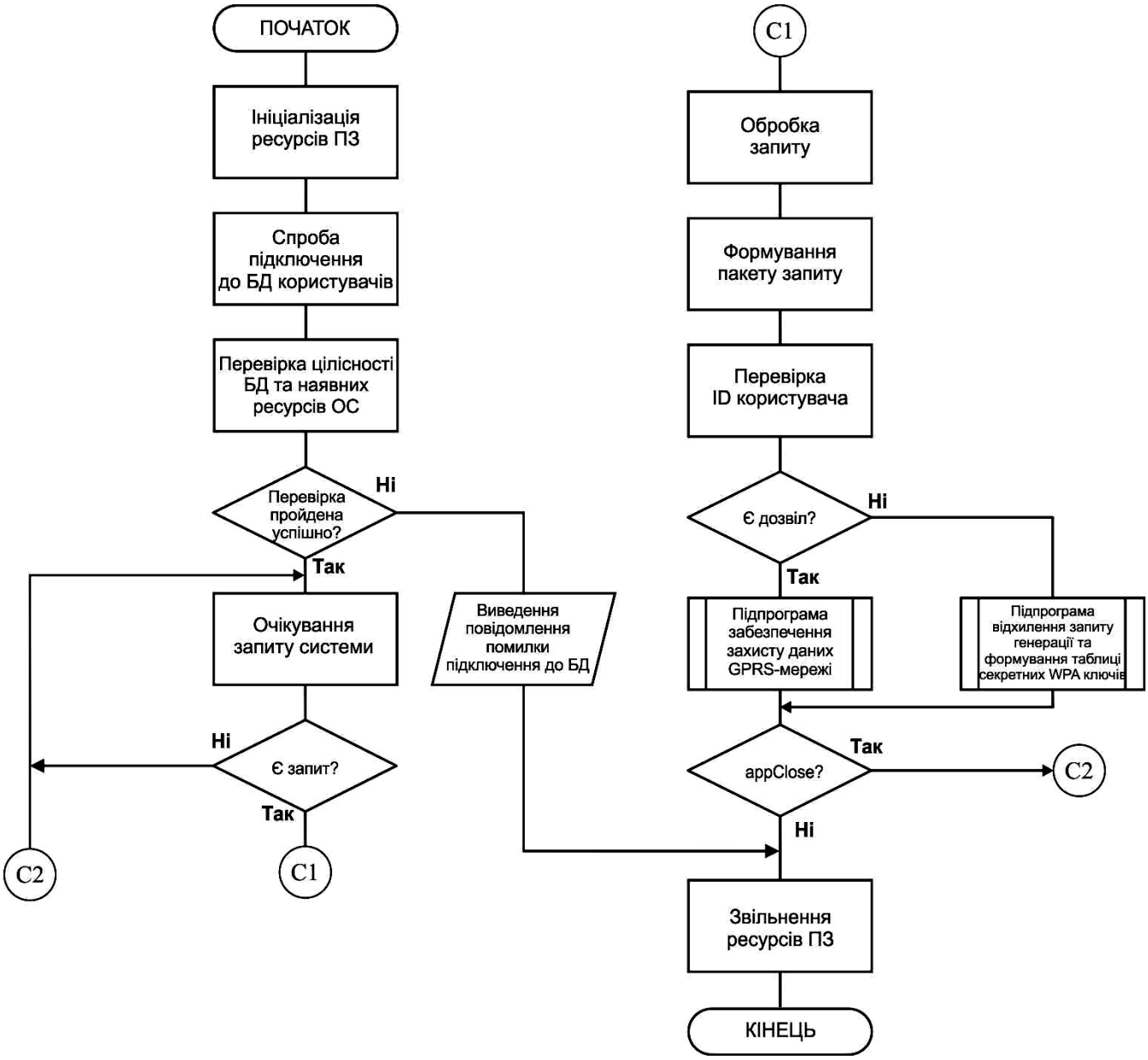

Рисунок 4.1 – Блок схема основної програми

Така організація дозволяє забезпечити незалежне виконання основних операцій і локалізувати можливі помилки без необхідності повного перезапуску мобільного пристрою.

Особливістю запропонованого рішення є виконання мережевого захисту засобами Android без обов'язкового модифікування ядра операційної системи. Для перехоплення маршрутизованих IP-пакетів може використовуватися VpnService, який створює віртуальний мережевий інтерфейс типу TUN.

Через нього програма отримує пакети, що відповідають налаштованим маршрутам і правилам VPN. Водночас цей механізм не надає прямого доступу до радіоінтерфейсу модема, алгоритмів шифрування стільникової мережі або службової сигналізації оператора.

Тому предметом захисту є насамперед IP-трафік пристрою, який передається через мобільне з'єднання. Втручання у функціонування базового процесора модема не передбачається.

Для детального розгляду процедур захисту доцільно виділити підпрограму контролю та маршрутизації мережевих пакетів. Вона перевіряє характеристики отриманого пакета, встановлює відповідність політиці безпеки та визначає спосіб його подальшого передавання.

Алгоритм функціонування цього модуля передбачено на рисунку 4.2.

Результатом оброблення є дозвіл на передавання через захищений тунель, відхилення пакета або реєстрація події для подальшого аналізу. Прийняття рішення повинно виконуватися до передавання даних у зовнішню мережу.

Обґрунтування архітектури захисту на рівні мережевої підсистеми Android

Операційна система Android використовує багаторівневу архітектуру, у якій прикладні процеси взаємодіють із мережевою підсистемою через системні інтерфейси. Для стільникового зв'язку важливими компонентами є служби керування підключеннями, підсистема телефонії та програмні інтерфейси взаємодії з модемом. Безпосередня реалізація радіопротоколів і значна частина операцій базового рівня виконуються спеціалізованим модемним обладнанням та його програмним забезпеченням. Звичайний Android-застосунок не має повноважень змінювати алгоритми шифрування радіоканалу або довільно перехоплювати службові повідомлення модема. Для таких операцій необхідні спеціальні системні привілеї, підтримка виробника обладнання та відповідні можливості прошивки. Тому проєктне рішення доцільно будувати на доступних механізмах Android Framework, не підміняючи ними функції фізичного та канального рівнів стільникової мережі.

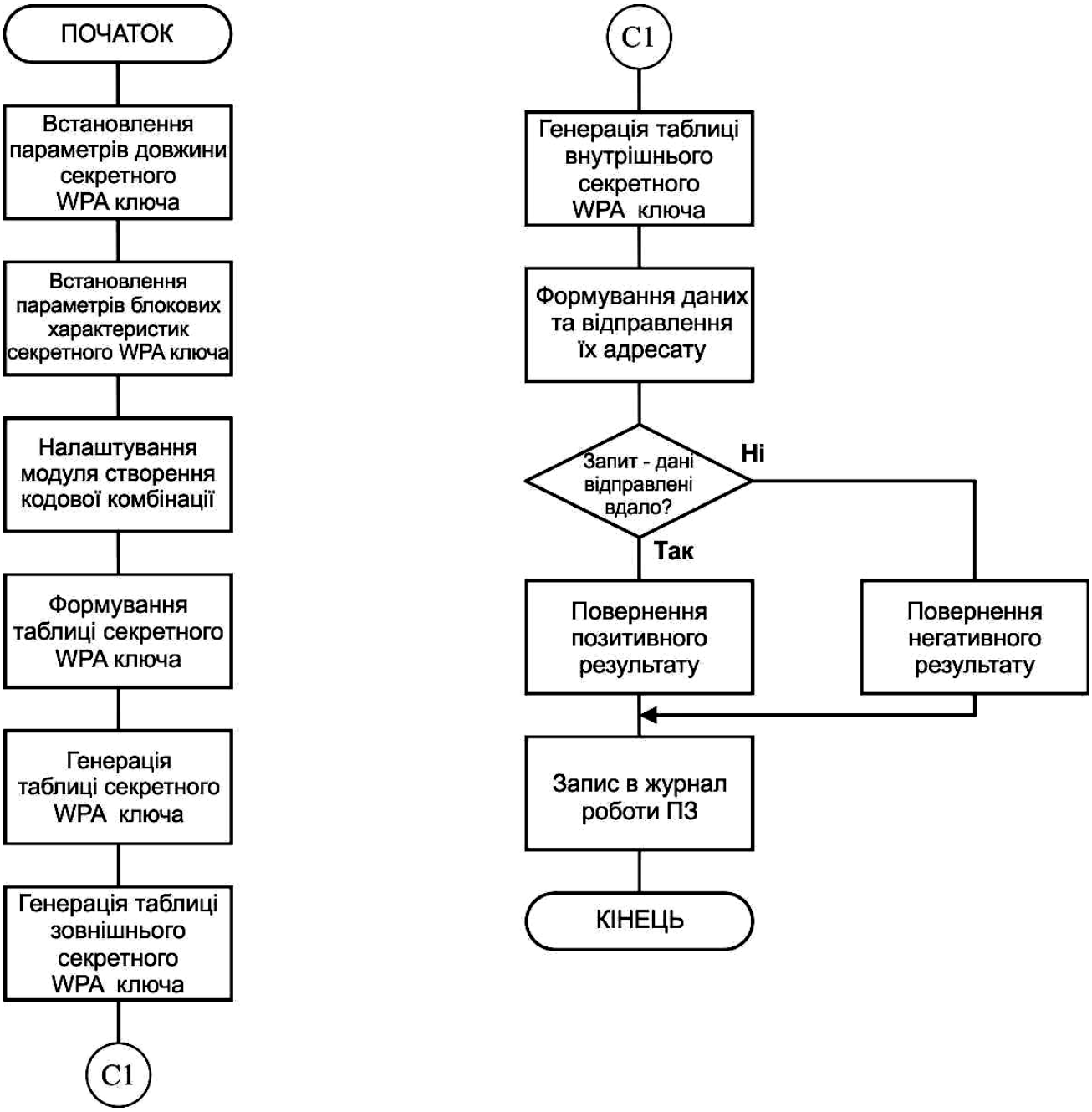

Рисунок 4.2 – Блок схема підпрограми

У межах запропонованої архітектури захищений канал формується між мобільним пристроєм і довіреним мережевим шлюзом. Шлюз розташовується у контрольованій серверній інфраструктурі та забезпечує приймання захищених пакетів, перевірку їхньої автентичності, розшифрування й подальшу маршрутизацію. На мобільному пристрої функціонує клієнтський компонент, який керує тунелем і передаванням даних через нього.

Апаратна складова комплексу може бути представлена засобами захищеного виконання криптографічних операцій мобільного пристрою. Зокрема, Android Keystore дозволяє створювати та використовувати ключі, які за наявності відповідного обладнання можуть бути пов'язані з Trusted Execution Environment або StrongBox. Наявність апаратного захисту необхідно перевіряти для конкретного ключа. Сам факт використання Android Keystore не гарантує його зберігання в захищеному апаратному середовищі.

Якщо вимоги до апаратного захисту передбачають використання зовнішнього криптографічного пристрою, архітектура може бути доповнена спеціалізованим модулем через USB. У такому випадку необхідно реалізувати окремий програмний адаптер для виклику підтримуваних криптографічних операцій. Зовнішній пристрій не повинен автоматично отримувати необмежений доступ до мережевого трафіку. Його функції доцільно обмежити автентифікацією, захищеним зберіганням ключів або виконанням визначених криптографічних операцій.

Для системи пропонується розділення функцій на площину керування та площину передавання даних. Площина керування забезпечує автентифікацію, отримання політик безпеки, керування життєвим циклом VPN-з'єднання й реєстрацію подій. Площина передавання відповідає за приймання IP-пакетів, перевірку правил маршрутизації та їх передавання через криптографічно захищений тунель.

Таке розмежування дозволяє змінювати параметри керування без втручання у внутрішню структуру пакетів прикладних протоколів. Наприклад, оновлення політики доступу може здійснюватися незалежно від роботи криптографічного модуля.

Водночас зміни, які впливають на маршрутизацію, повинні застосовуватися узгоджено, щоб уникнути тимчасового проходження трафіку поза захищеним каналом.

Важливим архітектурним обмеженням є відмінність між захистом IP-трафіку та захистом голосових викликів і повідомлень стільникової мережі.

Службові механізми телефонії, зокрема голосові послуги оператора та передавання SMS, не обов'язково використовують звичайний IP-тракт, доступний через VpnService. Тому запропонований комплекс не слід розглядати як універсальний засіб захисту всіх каналів стільникового зв'язку.

Для прикладних сервісів, що використовують IP-мережу, захищене передавання можна забезпечити незалежно від типу фізичного підключення. Це створює можливість використання одного механізму захисту під час роботи через LTE, 5G або Wi-Fi.

Однак дослідження системи в межах поставленої теми доцільно зосередити на стільниковому з'єднанні, оскільки для нього характерні зміни параметрів радіомережі, коливання затримок та переходи між базовими станціями.

Розроблення механізму ініціалізації та перевірки захищеного середовища

Початковий етап роботи комплексу передбачає перевірку можливості створення захищеного каналу. Для цього необхідно встановити доступність системних функцій Android, визначити стан мережевого підключення та перевірити наявність необхідних криптографічних засобів. Якщо хоча б одна обов'язкова умова не виконується, запуск передавання захищених даних повинен бути заблокований.

Алгоритм ініціалізації починається з отримання конфігураційних параметрів. До них належать адреса довіреного шлюзу, ідентифікатор пристрою, режим маршрутизації, перелік дозволених мережевих напрямків і вимоги до автентифікації. Параметри повинні проходити перевірку формату та допустимості значень.

Некоректна конфігурація не повинна передаватися наступним модулям.

Наступним кроком є перевірка стану криптографічного сховища. Якщо політика безпеки вимагає апаратного зберігання ключового матеріалу, програма повинна перевірити відповідний рівень захисту. За відсутності необхідних можливостей система може повідомити про несумісність пристрою. Автоматичне

використання менш захищеного сховища без окремого дозволу не відповідає вимогам суворої політики безпеки.

Для контролю запуску доцільно використовувати модель скінченного автомата. Основними станами можуть бути INITIALIZED, AUTHENTICATING, CONNECTING, PROTECTED, DEGRADED і BLOCKED. Кожен стан характеризує певний етап підготовки або виконання захищеного передавання.

Перехід від INITIALIZED до AUTHENTICATING здійснюється після перевірки конфігурації та доступності криптографічних ключів. Успішна автентифікація дозволяє перейти до встановлення тунелю. Після підтвердження захищеного з'єднання система переходить у стан PROTECTED. Якщо з'єднання втрачено, використовується стан DEGRADED. Стан BLOCKED означає, що необхідні умови захисту не виконано та передавання через комплекс не дозволяється.

Для формалізації правил переходу можна представити стан системи як

$$S_t \in \{S_{\text{init}}, S_{\text{auth}}, S_{\text{conn}}, S_{\text{prot}}, S_{\text{degr}}, S_{\text{block}}\}.$$

Перехід до захищеного стану допускається лише за виконання встановлених умов

$$S_{(t+1)} = S_{\text{prot}} \text{ за умови } A_t \wedge K_t \wedge T_t \wedge P_t,$$

де A_t означає успішну автентифікацію, K_t визначає доступність необхідних криптографічних ключів, T_t підтверджує готовність захищеного тунелю, а P_t характеризує коректність активної політики безпеки.

Для демонстрації принципу керування станами можна використати Python.

Наведений фрагмент ілюструє логіку програмного контролера, але не виконує безпосередніх системних викликів Android.

```
from dataclasses import dataclass
from enum import Enum, auto
class SecurityState(Enum):
    # Основні стани комплексу захисту
    INITIALIZED = auto()
    AUTHENTICATING = auto()
    CONNECTING = auto()
    PROTECTED = auto()
    DEGRADED = auto()
    BLOCKED = auto()
@dataclass
```

```
class SecurityContext:
    # Результати перевірки компонентів захисту
    authenticated: bool = False
    hardware_key_ready: bool = False
    tunnel_ready: bool = False
    policy_valid: bool = False

class SecurityController:
    def __init__(self):
        self.state = SecurityState.INITIALIZED
    def evaluate(self, context: SecurityContext):
        # Перевіряємо коректність політики безпеки
        if not context.policy_valid:
            self.state = SecurityState.BLOCKED
        # Контролюємо наявність апаратного ключа
        elif not context.hardware_key_ready:
            self.state = SecurityState.BLOCKED
        # Перевіряємо автентифікацію пристрою
        elif not context.authenticated:
            self.state = SecurityState.AUTHENTICATING
        # Очікуємо встановлення захищеного тунелю
        elif not context.tunnel_ready:
            self.state = SecurityState.CONNECTING
        else:
            # Усі необхідні умови виконано
            self.state = SecurityState.PROTECTED
        return self.state
    def on_connection_lost(self):
        # Фіксуємо втрату захищеного каналу
        if self.state == SecurityState.PROTECTED:
            self.state = SecurityState.DEGRADED
        return self.state
```

Запропонований контролер дозволяє визначити поточну готовність системи за результатами перевірки її компонентів. У повноцінній реалізації необхідно додатково перевіряти допустимість переходів між станами та забезпечити синхронізацію роботи контролера з фактичним станом VPN-служби.

Важливо враховувати, що встановлення внутрішнього стану BLOCKED саме по собі не забезпечує системного блокування мережевого трафіку інших застосунків.

Для цього необхідно використовувати відповідні механізми Android, зокрема Always-on VPN із забороною підключень поза VPN. Такий режим може налаштовуватися користувачем або адміністративними засобами керування пристроєм. У разі відсутності системного блокування застосунок не може гарантувати повне запобігання витоку трафіку під час втрати тунелю.

Реалізація контролю IP-трафіку та застосування політик безпеки

Після встановлення захищеного з'єднання основним завданням програмного комплексу є оброблення мережевих пакетів. Для цього використовується віртуальний інтерфейс TUN, який забезпечує обмін IP-пакетами між мережевою підсистемою Android і VPN-службою. Програмний компонент зчитує пакети з інтерфейсу, перевіряє їхні характеристики та передає дозволені дані захищеним каналом.

Для створення віртуального інтерфейсу VpnService використовує механізм VpnService.Builder. Він дозволяє визначати IP-адреси, маршрути та DNS-сервери VPN-з'єднання. Налаштування повинно враховувати одночасну підтримку IPv4 та IPv6.

Якщо захист передбачений для обох протоколів, відповідні маршрути необхідно встановити явно. Відсутність маршруту для одного з підтримуваних сімейств адрес може призвести до небажаного обходу захищеного каналу залежно від конфігурації системи.

Формально мережевий пакет можна представити набором характеристик

$$P_i = (a_s, a_d, p_s, p_d, \pi, l, t),$$

де a_s та a_d визначають адреси відправника й одержувача, p_s і p_d є номерами транспортних портів, π характеризує транспортний протокол, l визначає довжину пакета, а t є часом його надходження.

Не всі зазначені характеристики доступні для кожного IP-пакета. Наприклад, фрагментований пакет може не містити транспортного заголовка. Тому модуль аналізу повинен враховувати особливості IPv4, IPv6 та механізмів фрагментації.

Якщо необхідні поля відсутні або пакет має некоректний формат, рішення приймається відповідно до політики оброблення невизначеного трафіку.

Для системи захисту доцільно застосувати модель політик, засновану на наборі правил. Кожне правило визначає умови відповідності пакета та необхідну дію. Умови можуть охоплювати IP-адресу призначення, транспортний протокол,

порт і мережевий сегмент. Серед основних дій передбачаються дозвіл на передавання через тунель, блокування та реєстрація події.

Результат перевірки можна визначити функцією

$$D(P_i, R) = \begin{cases} \text{ALLOW}, & P_i \in R_{\text{allow}}, \\ \text{DENY}, & P_i \in R_{\text{deny}}, \\ \text{DEFAULT}, & \text{в інших випадках}, \end{cases}$$

де R є множиною правил, а DEFAULT визначає дію за відсутності відповідного правила. Якщо дозволи та заборони перетинаються, система повинна застосовувати заздалегідь визначену пріоритетність. Для захисного профілю доцільно встановити блокування за замовчуванням, коли жодне правило не підтверджує можливість передавання.

Порядок оброблення пакета передбачає перевірку коректності структури, отримання доступних параметрів, послідовне зіставлення з правилами й визначення рішення. Якщо пакет дозволено, він передається модулю захищеного транспорту. Заборонені пакети відкидаються, а відповідна подія може бути зареєстрована в журналі.

Приклад програмної реалізації перевірки мережевого призначення представлено мовою Python. Він демонструє модель фільтрації за IP-адресами, транспортними протоколами та портами. Отримання параметрів із реальних IP-пакетів має виконувати окремий парсер.

```
from dataclasses import dataclass
from ipaddress import ip_address, ip_network

@dataclass
class PacketInfo:
    # Параметри мережевого пакета
    destination: str
    protocol: str
    destination_port: int | None
@dataclass
class SecurityRule:
    # Правило оброблення мережевого трафіку
    network: str
    protocol: str
    port: int | None
    action: str
class TrafficPolicy:
    def __init__(self, rules):
        # Зберігаємо правила в заданому порядку
        self.rules = rules
    def evaluate(self, packet: PacketInfo):
```

```
try:
    # Перевіряємо адресу призначення
    destination = ip_address(packet.destination)
    # Перевіряємо допустимий транспортний протокол
    protocol = packet.protocol.upper()
    if protocol not in {"TCP", "UDP", "ICMP", "ICMPV6"}:
        return "DENY"
    # Перевіряємо транспортний порт
    if protocol in {"TCP", "UDP"}:
        if packet.destination_port is None:
            return "DENY"
        if not 0 <= packet.destination_port <= 65535:
            return "DENY"
    else:
        if packet.destination_port is not None:
            return "DENY"
    # Послідовно застосовуємо правила політики
    for rule in self.rules:
        network = ip_network(
            rule.network, strict=False
        )
        if destination not in network:
            continue
        if protocol != rule.protocol.upper():
            continue
        if rule.port is not None:
            if packet.destination_port != rule.port:
                continue
        # Повертаємо дозволену дію
        if rule.action in {"ALLOW", "DENY"}:
            return rule.action
        return "DENY"
except (ValueError, TypeError):
    # Некоректні параметри не допускаються
    return "DENY"
# Політика за замовчуванням
return "DENY"
```

Наведений алгоритм реалізує послідовне зіставлення пакета з правилами. Перше правило, яке відповідає його характеристикам, визначає результат. Такий підхід потребує впорядкування правил за пріоритетом. Для зменшення ймовірності помилок конфігурації доцільно виконувати перевірку суперечливих і дубльованих правил до їх застосування.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програмне забезпечення апаратно-програмного комплексу захисту каналів зв'язку стільникової мережі на системному рівні ОС Android призначене для контролю мережевого обміну, формування захищеного каналу передавання даних і керування політиками інформаційної безпеки мобільного пристрою.

Основними функціями програми є перевірка стану захищеного середовища, керування VPN-з'єднанням, фільтрація IP-трафіку та контроль доступності довіреного мережевого шлюзу. Головне вікно програми показано на рисунку 5.1.

Робоче середовище комплексу передбачає використання мобільного пристрою з ОС Android і серверного шлюзу, через який здійснюється захищене передавання даних. Програмний компонент мобільного пристрою забезпечує керування мережевими підключеннями та застосування правил безпеки.

Апаратна складова може використовувати захищене середовище виконання Trusted Execution Environment або модуль StrongBox за наявності їх підтримки пристроєм. Запропонована архітектура забезпечує захист IP-трафіку, який передається через стільникове з'єднання, без безпосереднього втручання у функціонування радіомодема.

Для реалізації мобільного застосунку доцільно використовувати Kotlin і стандартні програмні інтерфейси Android SDK. Основним компонентом захищеної маршрутизації є VpnService, який дозволяє створювати віртуальний мережевий інтерфейс TUN та отримувати IP-пакети.

Засоби Android Keystore використовуються для створення й застосування криптографічних ключів. Контроль доступності мережі організовується через ConnectivityManager. Алгоритми перевірки політик, керування станами та маршрутизації попередньо представлені мовою Python. Їх інтеграція з Android потребує відповідної реалізації у мобільному застосунку.

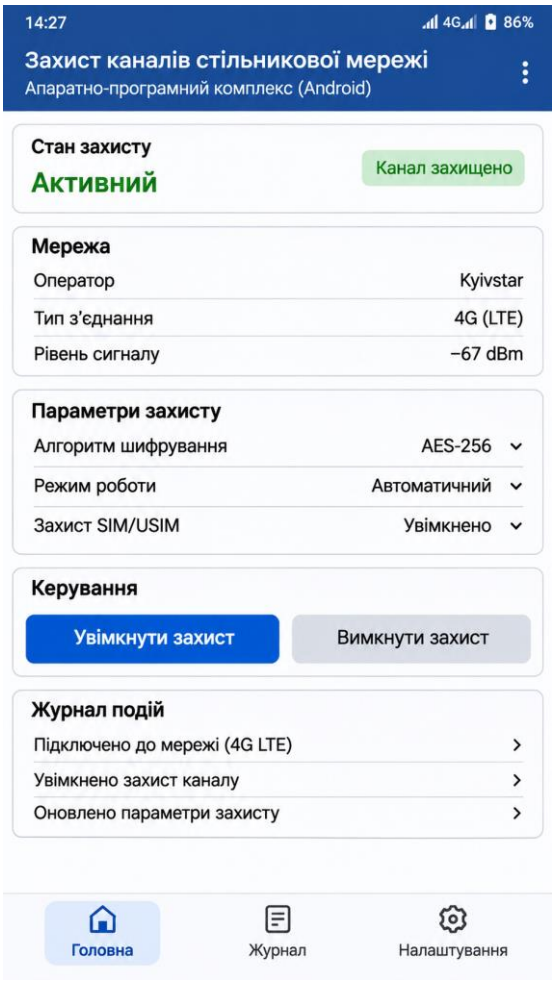

Рисунок 5.1 – Основне вікно програми

Для розгортання комплексу необхідний мобільний пристрій із підтримуваною версією Android, доступом до стільникового передавання даних і можливістю використання VPN-служби. Конкретний мінімальний рівень Android API визначається під час збирання застосунку.

Якщо політика передбачає апаратне зберігання ключів, необхідно перевірити відповідні можливості обладнання. Серверна частина повинна забезпечувати захищене мережеве з'єднання, автентифікацію клієнтів і маршрутизацію дозволеного трафіку. Характеристики сервера визначаються кількістю одночасних підключень та очікуваним навантаженням.

Підготовка програмного середовища починається зі встановлення Android Studio та необхідних компонентів Android SDK. Для створення застосунку

потрібно налаштувати проєкт, визначити мінімальну підтримувану версію операційної системи та підключити необхідні програмні залежності. У конфігурації застосунку передбачається оголошення VPN-служби з відповідними системними дозволами. Особливості її роботи у фоновому режимі повинні відповідати вимогам цільової версії Android. Після збирання застосунок встановлюється на тестовий пристрій для перевірки доступності системних функцій.

Наступним етапом є підготовка довіреного мережевого шлюзу. Для нього необхідно налаштувати мережевий інтерфейс, параметри захищеного транспортного протоколу та засоби автентифікації клієнтів. Як можливий варіант захищеного транспорту може розглядатися DTLS 1.3 за умови підтримки відповідною серверною та клієнтською реалізаціями. Конкретний протокол повинен забезпечувати конфіденційність, цілісність і захист від повторного відтворення повідомлень. Перед підключенням мобільних пристроїв потрібно перевірити правильність налаштування сертифікатів і доступність шлюзу через стільникову мережу.

Первинне налаштування мобільної частини передбачає визначення адреси довіреного шлюзу, параметрів автентифікації та правил маршрутизації. Для створення захищеного каналу необхідно сформувати конфігурацію віртуального інтерфейсу із зазначенням IP-адрес, маршрутів і DNS-серверів. Окремо враховується підтримка IPv4 та IPv6. Якщо політика передбачає передавання всього IP-трафіку через тунель, відповідні маршрути мають охоплювати обидва сімейства адрес. Усі конфігураційні параметри повинні перевірятися до активації VPN-з’єднання.

Для підготовки криптографічного середовища необхідно створити ключовий матеріал засобами Android Keystore. Якщо передбачено використання апаратно захищеного ключа, програма повинна перевірити фактичний рівень його захисту. Закриті ключі не повинні експортуватися до прикладного програмного середовища. Публічні ключі або відповідні сертифікати використовуються під час реєстрації пристрою на довіреному шлюзі. За відсутності необхідних апаратних

можливостей комплекс повинен повідомляти про неможливість виконання встановленої політики безпеки.

Рекомендована послідовність запуску починається з перевірки доступності стільникового підключення та завантаження конфігурації безпеки. Після цього виконується ініціалізація криптографічних засобів і перевірка облікових даних. Наступним кроком є отримання системного дозволу на створення VPN-з’єднання, якщо такий дозвіл ще не надано. Перед встановленням транспортного з’єднання із шлюзом необхідно виключити відповідний сокет із VPN-маршрутизації за допомогою VpnService.protect(). Після цього створюється віртуальний інтерфейс і встановлюється захищений сеанс.

Для запобігання передаванню даних поза захищеним каналом доцільно використовувати режим Always-on VPN із системним блокуванням підключень за відсутності VPN. Його активація здійснюється доступними засобами налаштування Android або адміністративного керування пристроєм. Звичайний застосунок не може самостійно гарантувати примусове застосування цього режиму. Перед введенням комплексу в експлуатацію необхідно перевірити фактичний стан системних налаштувань і відсутність незахищених маршрутів для контрольованого трафіку.

Після встановлення захищеного з’єднання активується модуль контролю мережевих пакетів. Він отримує дані з віртуального інтерфейсу, аналізує доступні параметри IP-пакетів та перевіряє відповідність установленим правилам. Дозволені пакети передаються захищеним каналом до мережевого шлюзу. Заборонені або некоректні пакети відхиляються. Для контролю роботи комплексу необхідно передбачити реєстрацію помилок маршрутизації, порушень політики та змін стану захищеного з’єднання.

Для оцінювання експлуатаційних характеристик рекомендується виконати вимірювання пропускної здатності, затримки передавання й використання обчислювальних ресурсів мобільного пристрою. Показники необхідно порівнювати для звичайного мережевого з’єднання та режиму із захищеним тунелем. Окремо доцільно визначати час відновлення після порушення зв’язку.

Отримані результати дозволяють оцінити накладні витрати програмно-апаратного захисту та встановити допустимі режими експлуатації.

Для ідентифікації програмного продукту й забезпечення його подальшого супроводу доцільно передбачити відображення відомостей про розробника, призначення програми та її версію. Така інформація дозволяє встановлювати відповідність установленого застосунку супровідній документації та контролювати оновлення програмних компонентів. Авторські дані розробленого програмного забезпечення апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі представлені на рисунку 5.2.

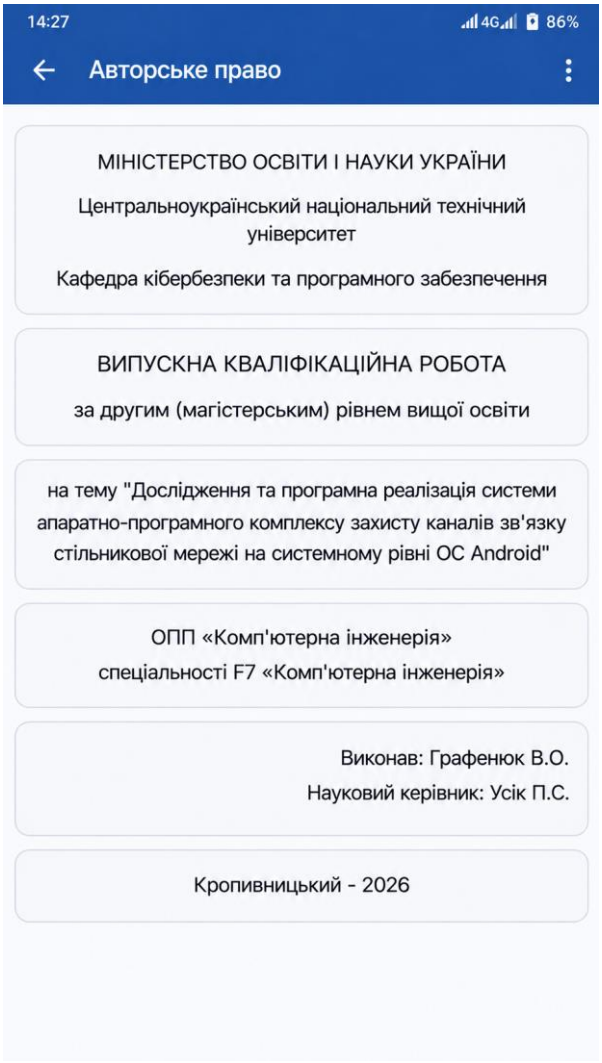

Рисунок 5.2 – Вікно авторського права

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

Метою розробки є дослідження та програмна реалізація системи апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

Об’єктом дослідження є процес апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

Предметом дослідження є методи апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

Методи дослідження базуються на методах теорії побудови комп’ютерних мереж, методах теорії захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

– Розроблено вітчизняний продукт апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android, який має більш широкі можливості, на відміну від існуючих аналогів.

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

7.1 Вступ

У сучасному житті електронно-обчислювальна машина (ЕОМ) займає важливе місце. Нині ми можемо уявити свою професійну діяльність без такого обов'язкового елемента нашого побуту. Зараз за допомогою ЕОМ керуються життєво важливі процеси. Це і управління різними електростанціями, аеропортами, постачання та логістика, документообіг, комерція та розробки, управління виробничими процесами тощо.

Але ЕОМ має і зворотний бік, а саме – небезпеку. Адже вона є складним електронним пристроєм, у якому є безліч процесів, які можуть стати джерелом небезпеки. З ЕОМ потрібно поводитися обережно та знати, як убезпечити себе від негативних наслідків його використання. Оскільки захворювання можуть бути спричинені надмірним фізичним або розумовим навантаженням, через велику нервово-емоційну напругу, або через виробниче середовище. Також програмісти у процесі роботи отримують негативний вплив на органи зору та руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження. Ці шкідливі фактори можуть привести до професійних захворювань. В даному розділі магістерської роботи проведемо аналіз основних чинників при роботі програміста.

Законом України “Про охорону праці” [7] регламентуються загальні положення державної політики в галузі охорони праці, а також конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [8], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час

експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-97.

Законодавством України чітко врегульовано норми та вимоги до використання комп’ютерної техніки на підприємстві, безпосередньо й охорона праці на підприємстві при роботі за комп’ютером., зокрема «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями», затверджені наказом Мінсоцполітики від 14.02.2018 № 207 [1], «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-97. [2].

Загальні вимоги пожежної безпеки під час експлуатації комп’ютерної техніки визначають «Правила пожежної безпеки в Україні» (затверджені наказом МВС від 30.12.2014 № 1417) [3], комп’ютерних класів – пункт 3 розділу VIII «Правил пожежної безпеки для навчальних закладів та установ системи освіти України» (затверджені наказом МОН від 15.07.2016 № 974). [4] та інші державні стандарти, що регламентують експлуатування комп’ютерної техніки як радіоелектронної апаратури.

7.2 Шкідливі і небезпечні фактори при роботі з комп’ютером

Можна виділити наступні основні фактори, що впливають на стан здоров'я людей, які працюють за комп'ютером:

– сидяче положення на протязі тривалого періоду;

– вплив електромагнітного випромінювання монітора;

– втома очей, навантаження на зір;

– перевантаження суглобів кистей;

– стрес при втраті інформації або виникненні критичних помилкою.

У кожному з цих випадків ступінь ризику прямо пропорційний часу, що проводиться за комп’ютером і поблизу від нього. В сучасних умовах взаємодія людини з технікою значно ускладнилась, що вимагає комплексного підходу,

який передбачає розгляд людини, технічних засобів праці та виробничого середовища, як взаємозв'язаних елементів єдиної системи. Все вищесказане в повній мірі відноситься й до системи «людина–комп'ютер–середовище».

Вагомий вплив на працездатність та здоров’я користувачів комп’ютерів здійснює виробниче середовище. Це середовище у виробничих приміщеннях (офісах), в основному, визначається мікрокліматом, освітленням, наявністю шкідливих речовин у повітрі, рівнем шуму та випромінювання.

Для того, щоб об’єктивно проаналізувати відповідність умов праці діючим нормативно-правовим актам та запропонувати заходи щодо зменшення негативного впливу комп’ютера на організм людини необхідно скласти санітарно-гігієнічну характеристику умов працівника, який працює з програмним продуктом.

7.3 Аналіз санітарно-гігієнічних умов праці на робочому місці користувача ПК

Розглянемо приміщення в якому працює користувач ПК з даним програмним продуктом.

Приміщення має одностороннє природне освітлення і загальне штучне освітлення. Стіни і стеля обклеєні світлими шпалерами, підлога вкрита темним ламінатом. У приміщенні відсутні сильні вібрації та шкідливі речовини. Склад повітря відповідає нормі. У кімнаті знаходиться ПК з 4-ядерним процесором і 23-дюймовим IPS-монітором, а також меблі.

Приміщення має довжину 4 м, ширину 3,5 м, висоту стелі 2,7 м. Кількість робочих місць – одне. Площа – 14 м^2, об'єм – 37,8 м^3. Виходячи з цього, отримано дані, наведені в таблиці 7.1.

Таблиця 7.1 – Фактичні та нормативні значення параметрів приміщення

Параметр	Норма *	Реальні параметри
Площа, S	не менше 6 м^2	14 м^2
Об’єм, V	не менше 20 м^3	37,8 м^3

*Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

За даними, які наведено у табл. 7.1, можна зробити висновок, що отримані показники, площа та об’єм приміщення у розрахунку на одно робоче місце користувача ПК відповідає чинним нормам і вимогам.

Щодо мікроклімату, то згідно з ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» [5] роботу з ПК можна віднести до категорії легка – 1а. Джерелами тепла в цьому приміщенні є люди, електроустаткування, освітлювальні прилади в темний час доби і система опалювання взимку. Оператором виділяється до 120 ккал теплової енергії за годину. Оптимальні та фактичні значення параметрів мікроклімату приведені в таблиці 7.2.

Таблиця 7.2 – Значення параметрів мікроклімату

Період року	Параметр	Оптимальний*	Фактичний
Теплий	Температура	23 – 25 ^{0} С	24^0 С
	Вологість	40 – 60%	50%
	Швидкість повітря	< 0,1м/с	
Холодний	Температура	22 – 24^0 С	23^0 С
	Вологість	40 – 60%	55%
	Швидкість повітря	< 0,1м/с	

*ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»

По отриманим замірам параметрів мікроклімату можна зробити висновок, що всі показники задовольняють вимогам, зазначеним для робіт категорії легка 1а і є задовільними для здоров'я людини. Щодо освітлення, то згідно з ДБН В.2.5-28:2006 «Природне і штучне освітлення» [6] ця робота

відноситься до Va розряду зорових робіт. Передбачається використання природного, штучного і змішаного освітлення.

Природне освітлення здійснюється за допомогою вікна, площа якого складає S' = 1,8х1,5 = 2,7 м^2 і є бічним освітленням. У світильниках місцевого і загального освітлення використовуються світлодіодні лампи потужністю 20 Вт із світловим потоком однієї лампи 900 лм. Згідно замірів рівень освітлення в даному приміщенні і на робочому місці складає в межах 350 -500 лк, що відповідає нормованому значенню.

Джерелом шуму в приміщенні є комп'ютер. Вентилятори (кулери) системного блоку, процесора, відеокарти і блоку живлення є сучасними і мають низький рівень шуму. Згідно з технічною документацією шум, зумовлений кулером в блоці живлення складає 25 дБ, кулером процесора – 30 дБ, загальний – 34 дБ. Враховуючи незначний рівень шуму від персонального комп'ютера і незначний рівень фонового шуму від іншого устаткування, можна стверджувати, що сумарний рівень шумового забруднення приміщення не перевищує максимально допустимий рівень коригованої звукової потужності і складає не більше 50 дБА, що відповідає рівню шуму для приміщень з комп’ютерною технікою згідно Державних санітарних правил і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-97.

У приміщенні відсутні джерела інфрачервоного, ультрафіолетового і електромагнітного випромінювання, бо монітор ПК вироблений на основі рідкокристалічної матриці, підсвітка якої здійснюється неоновою лампою, які не мають сильного електромагнітного випромінювання і сертифіковані в Україні.

Блок живлення є екранованим і не випускає вищезазначених видів випромінювання.

7.4 Розробка заходів з умов поліпшення охорони праці

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці користувача ПК.

З точки зору забезпечення електробезпеки до цих заходів можна віднести: устаткування розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв; періодична перевірка всіх приладів і пристроїв; щорічна здача іспитів з охорони праці.

З точки зору забезпечення оптимальних умов мікроклімату, рівня звуку і освітленості до цих заходів можна віднести: організацію природної вентиляції, за допомогою дефлектора, для забезпечення необхідного повітрообміну в приміщенні вузла; організацію системи центрального опалювання, для підтримки оптимальної температури в холодний період року; організацію штучного загального освітлення, для забезпечення необхідних умов зорової роботи, що відповідають, оформлення паспорта на приміщення вузла, з занесенням в нього вимірювань освітленості і рівня звуку, проведених відділом охорони праці.

Крім рекомендацій щодо конкретного приміщення, де було проведено дослідження умов праці, існують загальні вимоги, які зарекомендовані відповідними нормативними документами.

Правильна організація робочих місць запобігає передчасній втомлюваності користувача і сприяє збереженню здоров'я. Організація робочого місця передбачає:

– правильне розміщення робочого місця у виробничому приміщенні;

– вибір ергономічного обґрунтованого робочого положення, виробничих меблів з урахуванням характеристик людини;.

– раціональне компонування обладнання на робочих місцях;

– урахування характеру й особливостей трудової діяльності.

Стосовно робочих місць користувача ВДТ, то організація робочого місця має забезпечуватися відповідно до ДСанПіН 3.3.2-007-97. Для запобігання перевтомленню необхідно виконувати вправи для очей та дотримуватись розпорядку роботи та відпочинку. На робочому місці реалізовувався режим відпочинку: кожні дві години – перерва для виконання фізичних вправ для м‘язів очей.

7.5 Протипожежний захист

Пожежі в приміщеннях з оргтехнікою становлять особливу небезпеку, бо поєднані з великими матеріальними збитками. Пожежа може виникнути при взаємодії горючих речовин і джерел запалювання. Горючими речовинами є будівельні та опоряджувальні матеріали, пластмасові корпуси техніки, шнури тощо. Джерелами запалювання можуть бути електронні схеми комп'ютерів, принтерів, пристроїв електроживлення, де внаслідок різних порушень виникає перегрівання елементів, утворюються електричні іскри та дуги, здатні спричинити займання горючих матеріалів.

При обслуговуванні, ремонтних та профілактичних роботах використовуються різні легкозаймисті рідини, прокладаються тимчасові електропровідники, здійснюється паяння. Виникає додаткова пожежна небезпека, яка потребує відповідних заходів пожежного захисту.

До засобів гасіння пожежі, призначених для локалізації невеликих займань, належать вогнегасники, сухий пісок, азбестові ковдри. Приміщення, в якому встановлено комп'ютери і де немає необхідності влаштування систем автоматичного пожежогасіння, необхідно оснащувати переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 $м^2$ в приміщеннях.

Звуковбирне облицювання стін, стель приміщень треба виконувати з негорючих та важко горючих матеріалів.

З метою виявлення початкової стадії займання необхідно використовувати пристрої систем автоматичного пожежогасіння там, де цього вимагають Правила пожежної безпеки.

З точки зору забезпечення пожежної безпеки до цих заходів можна віднести наявність схеми евакуації з приміщення у випадку пожежі, прикріплену на вхідні двері.

7.6 Розрахункова частина

В приміщенні (де відсутні джерела виділення шкідливих речовин) працює двоє людей. Робота пов'язана з використанням ПЕОМ. Розміри приміщення: А = 4 м, В = 7 м, Н = 2.8 м, устаткування займає 15% об'єму. Визначимо найменшу необхідну кількість повітря для вентиляції.

Для приміщень, в яких відсутні виділення шкідливих речовин у повітрі, розрахунок вентиляції здійснюється залежно від кількості працюючих.

Необхідна кількість повітря ($м^3$/год.), яка забезпечує відповідність параметрів повітря робочої зони нормованим значенням, визначається за наступною формулою:

$$L = L' \cdot N,$$

де

L' – нормативна кількість повітря на одного працюючого, яка залежить від питомого об'єму приміщення, $м^3$/(год.-люд.);

N – кількість працюючих.

Питомий об'єм приміщення $V_п$, ($м^3$/люд), визначається за формулою:

$$V_n = V/N,$$

де V – об'єм приміщення, $м^3$.

Визначаємо вільний об'єм приміщення

$$V = A \cdot B \cdot H \cdot 0{,}85 = 4 \cdot 7 \cdot 2{,}8 \cdot 0{,}85 = 66{,}4 \text{ м}^3.$$

Питомий вільний об'єм складає

$$V' = V / N = 66{,}4 / 2 = 33{,}2 \text{ м}^3/\text{люд.} > 20 \text{ м}^3/\text{люд.}$$

Нормована кількість повітря на одну людину при V' > 20 м^3 /люд. становить 30 м^3 /(год.·люд.).

Висновки до розділу

У даному розділі магістерської роботи проведено аналіз умов працівника робота якого пов’язана з комп’ютерною технікою. Проведено аналіз основних санітарно – гігієнічних показників в заданому приміщені , де працівник зайнятий постійною роботою за комп’ютером. Створені умови повинні забезпечувати комфортну роботу. На підставі вивченої літератури з даної проблеми, були зазначені оптимальні параметри мікроклімату, освітлення, допустимі рівні шуму та іонізуючого випромінювання при роботі з ПЕОМ, а також розраховано найменшу необхідну кількість повітря для вентиляції.

Дотримання умов, що визначають оптимальну організацію робочих місць працівників, дозволить зберегти гарну працездатність протягом усього робочого дня, підвищить як в кількісному, так і в якісному відносинах продуктивність їх праці.

8 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

– Досліджена система апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

– На основі отриманих результатів досліджень створена програмна реалізація системи апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

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

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Android.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

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

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Графенюк В.О. Дослідження та програмна реалізація системи апаратно-програмного комплексу захисту каналів зв’язку стільникової мережі на системному рівні ОС Android // Збірник праць молодих науковців ЦНТУ. – Вип. 17. – Кропивницький: ЦНТУ, 2027.
2. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.
3. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.
4. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.
5. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p
6. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.
7. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.
8. Богаченко, С., Смірнов, О., Буравченко, К., Смірнова, Т., Конопліцька-Слободенюк, О., Якименко, Н., Смірнов, С. «Обмеження ресурсів мобільних та edge-платформ при розробці програмного забезпечення для реалізації та оптимізації моделей машинного навчання». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2026, т. 2 № 34, С. 677-686.
9. Вінтенко Б.Ю., Смірнова Т.В., Миронець І.В., Смірнов О.А., Буравченко К.О. «Метод оцінки функціональної стійкості комп'ютерно-орієнтованих процедур системи підтримки оперативного персоналу АЕС». *Системи управління, навігації та зв’язку*, 2026, вип. 2(84), С. 62-69.
10. Смірнова Т.В., Усік П.С., Лисенко І.А., Буравченко К.О., Смірнов О.А. «Методологія підтримки технологічних процесів у критичній інфраструктурі з забезпеченням безпеки інформації на основі хмарних

технологій». *Безпека інформації.* 2025. Том 31 № 1. С. 49-60.

11. Смірнова Т.В., Буравченко К.О., Добринчук О.А., Смірнов С.А., Якименко Н.М., Смірнов О.А. «Хмарна технологія моніторингу ключових індикаторів ефективності технологічних процесів у критичній інфраструктурі». *Центральноукраїнський науковий вісник. Технічні науки.* 2025. № 12(43), ч. II. С. 36-54.

12. Вінтенко, Б.Ю., Миронець, І.В., Смірнов, О.А., Коваленко, О.В., Усік, П.С., Буравченко, К.О., Лисенко, І.А. «Логіко-структурна модель комп'ютерно-орієнтованої процедури системи підтримки оперативного персоналу АЕС». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С. 413-427.

13. Vintenko, B., Myronets, I., Smirnov, O., Smirnova, T. «Enhancing the Reliability of Input Data for Nuclear Power Plant Operator Support Systems». *International Conference on Sensors, Devices and Systems. SDaS 2025. Lecture Notes in Electrical Engineering,* 2026. vol 1617. pp 3-12/

14. Kuznetsov O., Frontoni E., Arnesano M., Smirnov O., Khruskov B. «Biometric Authentication in TinyML: Opportunities and Challenges». *Tiny Machine Learning: Design Principles and Applications,* 2026, pp. 587-633.

15. Kuznetsov O., Frontoni E., Kuznetsova Y., Chevardin V., Smirnov O. «Architectural foundations for adaptive security in edge computing systems». *Cybersecurity Defensive Walls in Edge Computing*, 2025. pp. 21-61.

16. Chulinda L., Smirnov O., Shapenko L., Ustynova I., Bohatiuk I., Kelyp S. «The role of innovation in ensuring the safety of international civil aviation». *Ceur Workshop Proceedings,* 2025, 4024, pp. 530–542.

17. Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед та ін. «Системи інформаційної зброї та технології інформаційної війни»: підручник / Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед, Жарков Я.М., Смірнов О.А, Буравченко К.О., Давидюк А.В., Кононович В.Г., Корчинский В.В., Кудирко В.М., Фесенко А.О.; за заг. ред. В.М. Петрика, М.М. Присяжнюка.– К.: Видавничий центр “Кафедра”, 2025.– 320 с.

18. Усік, П.С., Смірнова, Т.В., Буравченко, К.О., Смірнов, О.А., Улічев,

О.С., Смірнов, С.А. «Дослідження технологій забезпечення кібербезпеки банківських систем з використанням штучного інтелекту». *Кібербезпека: освіта, наука, техніка*. 2025. Том 1 № 29. С.704–716, 2025

19. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 193–224.

20. Kuznetsov, O., Atzeni, G., Arnesano, M., Randieri, C., Smirnov, O. «Secure IoT-based smart wheelchair system: From implementation to security enhancement strategy». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 225–257.

21. Kuznetsov, O., Smirnov, O., Kuznetsova, T., Shaikhanova, A., Svatowsky, I. «Privacy-utility trade-offs in IoT networks: A comparative analysis of differential privacy mechanisms for sensor data aggregation». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 589–622.

22. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». *8th International Symposium on Intelligent Informatics, ISI 2023*, 2025. vol 389. pp 377-389. Springer, Singapore.

23. Kuznetsov O., Frontoni E., Kuznetsova Y., Smirnov O., Moskovchenko I. «Trust-Based Security Architecture for Edge Computing: A Simulation Study of Dynamic Trust Evolution and Attack Detection». *CEUR Workshop Proceedings*, 2024, 3909, pp. 227–241.

24. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.

25. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L.

«Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447.

26. Ткаченко, О., Ільєнко, А., Улічев, О., Мелешко, Є., Смірнов, О. «Правові засади поширення інформаційних впливів в соціальних мережах». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2024. № 2(26), С. 170–188.

27. Смірнова Т.В., Конопліцька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка.* 2024. №4(24), С. 6-27.

28. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка.* 2024. №3(23), С. 111-131.

29. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології,* 2024, № 13, с. 28-35.

30. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

31. Akhalaia, G., Iavich, M., Iashvili, G., Prysiazhnyy, D., Smirnova, T. «Secure Encrypted Connection on Georgian Website». *CEUR Workshop Proceedings*, 2023, 3550, pp. 313-320.

32. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56

33. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures».

CEUR Workshop Proceedings, Volume 3530, 2023, pp. 256-265.

34. Kuznetsov, O., Kandiy, S., Frontoni, E., Smirnov, O. «Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece». *CEUR Workshop Proceedings*, Volume 3504, 2023, pp. 1-11.

35. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». *In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. Springer, Singapore. pp. 21-34.

36. Смірнова Т.В., Гнатюк С.О., Бердибаєв Р.Ш., Сидоренко В.М., Жигаревич О.К., «Система корелювання подій та управління інцидентами кібербезпеки на об'єктах критичної інфраструктури». *Кібербезпека: освіта, наука, техніка*, №3(19), 2023, С. 176-196.

37. Смірнов О.А. Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *II Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи hозвитку інформаційних технологій (ІПШРІТ-2023)»* м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

38. Козлов Я.О., Смірнова Т.В., Смірнов О.А. «Дослідження SIEM-систем для забезпечення кібербезпеки». *VII міжнародна науково-практична конференція "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя кафедри кібербезпеки та програмного забезпечення,* м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 26.

39. Козлов Я.О., Козірова Н.Л., Смірнов О.А. «Дослідження структури та принципу роботи SIEM-системи». *VII міжнародна науково-практична конференція "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя кафедри кібербезпеки та програмного забезпечення,* м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 59.

40. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А.,

Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

41. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings,* Volume 3187, 2022,

42. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477.

43. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління,* № 2(70). 2022. С. 28-37.

44. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку,* 2022, № 3(69). С. 93-98.

45. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»,* № 2 (307). С. 46-52. 2022.

46. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

47. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021*, Cracow, Poland, 22-25 September 2021. P. 414-418

48. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». *4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) – 2021*, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

49. Smirnov O., Kuznetsov A., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». *2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020,* Kharkiv, 6 October 2020-9 October 2020, P. 358-362.

50. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». *CEUR Workshop Proceedings* Volume 2805, 2020, Pages *44-58.*

51. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». *International Journal of Computing*; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

52. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

53. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

54. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131.

55. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 1-14.

56. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. Springer, Cham. 2021, pp 66-84.

57. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

58. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I.,Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 166-171.

59. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

60. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

61. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

62. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». *International Journal of Computer Network and Information Security (IJCNIS)*. Vol. 12, No. 3, 2020. PP.33-43.