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 програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем”

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

Керівник проекту
доктор філософії (PhD), доцент
_____ Дрєєва Г.М.
«____» _____ 2026 р.

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

Рецензент _____
_____

Підписано КЕП
Центральноукраїнський національний технічний університет
Волошин Єгор Олександрович
Сертифікат: 18DF2C9B93CEA2040400000009E965E00AD4A5106
Дійсний з 21.10.2026 14:42:21 по 21.10.2027 14:42:20

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

Волошину Єгору Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем*

2. Керівник роботи *Дрєєва Ганна Миколаївна, доктор філософії (PhD), доцент*
(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 програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем*

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

Волошин Є.О. Дослідження та програмна реалізація системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем. 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 програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

Метою розробки є дослідження та програмна реалізація системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

Об'єктом дослідження є процес програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

Предметом дослідження є методи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

Методи дослідження базуються на методах схемотехніки, методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

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, програмно-апаратного скремблювання, цифрові сигнали, мобільні операційні системи

ABSTRACT

Voloshyn Ye.O. Research and Software Implementation of a Hardware-Software Digital Signal Scrambling System Based on Mobile Operating Systems. F7 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

This master's-level graduation thesis presents the development of software designed for a hardware-software digital signal scrambling system based on mobile operating systems.

The objective of the project is to research and implement a hardware-software digital signal scrambling system based on mobile operating systems.

The object of the study is the process of hardware-software digital signal scrambling based on mobile operating systems.

The subject of the study comprises the methods of hardware-software digital signal scrambling based on mobile operating systems.

The research methodology relies on circuit design techniques, information security methods, mathematical statistics, and software development practices.

The outcome of the work is the software implementation of a hardware-software digital signal scrambling system based on mobile operating systems.

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, and operating instructions are provided.

The software is designed for use on mobile devices running the Android operating system.

The software was developed using the Python environment.

Keywords: computer engineering, hardware-software scrambling, digital signals, mobile operating systems

ВІДОМІСТЬ ПРОЕКТУ

		Текстові документи			
1	А4	Пояснювальна записка	76	–	
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 12
1.1 Призначення системи..... 12
1.2 Область застосування 13
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 15
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... 15
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування 23
2.3 Розгорнута постановка завдання 23
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 25
3.1 Опис функціонування системи 25
3.2 Розробка структурної схеми..... 28
3.3 Розробка функціональної схеми 37
3.4 Розробка діаграми процесів 40
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ 42
4.1 Розробка блок-схем та опис алгоритмів функціонування системи 42
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 51
6 НАУКОВА НОВИЗНА 56

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ 57
7.1 Вступ 57
7.2 Характеристика умов праці програміста 58
7.3 Аналіз санітарно-гігієнічних умов праці
на робочому місці користувача ПК 60
7.4 Розробка заходів з охорони праці 62
7.5 Розрахункова частина 63
8 ОСНОВНІ ВИСНОВКИ 67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БД	–	база даних
ПЗ	–	програмне забезпечення
AES	–	Advanced Encryption Standard
USB	–	universal serial bus

ВСТУП

Актуальність теми. Комп'ютерна мережа призначена для передачі інформації з однієї точки в іншу. Дані, які ми надсилаємо, можуть бути цифровими або аналоговими. Також сигнали, що представляють дані, також можуть бути цифровими або аналоговими. Таким чином, щоб надсилати дані за допомогою сигналів, ми повинні мати можливість перетворювати дані на сигнали, це перетворення може бути аналогово-аналоговим, аналогово-цифровим, цифрово-аналоговим або цифро-цифровим. Цифро-цифрове перетворення включає три методи: лінійне кодування, блокове кодування та скремблювання. Лінійне кодування завжди потрібне, тоді як блокове кодування та скремблювання можуть бути потрібними або ні залежно від потреби. Скремблювання – це техніка, яка не збільшує кількість бітів і забезпечує синхронізацію. Проблема з такими методами, як біполярний AMI (альтернативна інверсія міток), полягає в тому, що безперервна послідовність нулів створює проблеми синхронізації, одним із рішень цієї проблеми є скремблювання. У даній роботі скремблювання призначено для захисту мобільних телефонів від прослуховування.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

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

– Огляд існуючих систем програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

– Дослідження системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

– Програмна реалізація системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

Об'єктом дослідження є процес програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

Предметом дослідження є методи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

Методи дослідження базуються на методах схемотехніки, методах 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 програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

– Розроблено вітчизняний продукт програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

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 статті збірника праць молодих науковців ЦНТУ, випуск №17.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем, 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

Мобільні телефони є важливими для спілкування, але їхні мікрофони також можуть становити загрозу безпеці. Навіть у режимі очікування смартфони можуть непомітно записувати та передавати конфіденційні розмови, розкриваючи конфіденційні дискусії для сторонніх слухачів.

Приховані ризики мікрофонів смартфонів

1. Несанкціонований запис

Телефони можуть бути залишені піднятими (активний дзвінок) навмисно або випадково, передаючи аудіо в реальному часі без вашого відома.

Шкідливі програми або шпигунські програми можуть таємно активувати мікрофон, записуючи розмови у фоновому режимі.

2. Зламані або скомпрометовані додатки

Багато програм (месенджери, соціальні мережі, голосові помічники) мають доступ до мікрофона – якщо їх зламати, вони можуть перетворитися на інструменти для підслуховування.

Навіть перевірені програми можуть мати вразливості, які зловмисники використовують для прослуховування.

3. Недостатній захист лише від глушіння

Хоча бездротові глушилки блокують передачу, вони не зупиняють запис – аудіо все ще можна зберігати на пристрої та отримувати пізніше.

Рішення: повна акустична блокування

Єдиний спосіб гарантувати конфіденційність – це фізично заблокувати мікрофони телефону від вловлювання звуку. Це забезпечує:

– Без прослуховування в реальному часі – зупиняє віддалене прослуховування в режимі реального часу.

– Немає збережених записів – Запобігає збереженню аудіо на пристрої.

– Не залежить від безпеки програмного забезпечення – працює навіть якщо телефон заражений шпигунським програмним забезпеченням.

Для важливих зустрічей, конфіденційних переговорів або безпечних дискусій телефонна будка із захистом від прослуховування – це найнадійніший спосіб запобігти витоку аудіо, зберігаючи ваші розмови справді конфіденційними.

1.2 Область застосування

Потенційні клієнти , яким може знадобитися захисний блок із захистом від прослуховування, щоб зупинити прослуховування телефонного мікрофона.

Ключові цільові клієнти:

1. Керівники компаній та члени правління

– Необхідність захисту стратегічних обговорень та чутливих процесів прийняття рішень.

– Регулярно брати участь у зустрічах, що стосуються нерозкритих фінансових даних та ринково-чутливої інформації.

2. Юридичні фірми та юридичні фахівці

– Обробляти конфіденційну інформацію клієнта та деталі справи.

– Повинен забезпечити захист адвокатської таємниці в умовах конференц-залу.

3. Команди досліджень та розробок та відділи інновацій

– Обговоріть передпатентні інновації та інтелектуальну власність.

– Вимагати захисту від збору конкурентної розвідки та корпоративного шпигунства.

4. Державні установи та посадовці

– Обробляти секретну, але не надсекретну інформацію.

– Необхідність балансу між вимогами прозорості та захистом конфіденційної інформації.

5. Фахівці з фінансових послуг

– Повинні відповідати вимогам фінансового регулювання.

– Обговоріть інвестиційні стратегії та конфіденційну фінансову інформацію клієнта.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем, 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

Corrata

Corrata забезпечує виявлення загроз та реагування на них, а також запобігання втраті даних для мобільних кінцевих точок на iOS та Android. Вона використовує глибоку перевірку пакетів на пристрої для перевірки трафіку на всіх портах та протоколах, забезпечуючи видимість загроз у режимі реального часу.



Рисунок 2.1 – Інтерфейс користувача Corrata

Він блокує фішингові атаки на кожному каналі, виявляє шпигунські програми та відключає їх зв'язок, контролює використання тіньового штучного інтелекту, запобігає втраті даних, захищає від перехоплення з боку зловмисника , відстежує конфігурацію пристроїв на наявність вразливостей та захищає скомпрометовані пристрої від конфіденційних даних.

Особливості:

1. Глибока перевірка пакетів на пристрої: перевіряє трафік програм на пристрої за всіма портами та ключовими протоколами, а не лише за DNS, з блокуванням IP-адрес та портів і динамічним виявленням шкідливого програмного забезпечення.

2. Захист від мобільного фішингу, що не обмежується електронною поштою: виявляє атаки, що надходять через сповіщення, текстові повідомлення, QR-коди та публікації в соціальних мережах, включаючи збор облікових даних, обхід багатофакторної автентифікації та крадіжку файлів cookie для автентифікації.

3. Тіньове управління ШІ та DLP: моніторинг порушень політики щодо підключень до споживчих послуг ШІ, несанкціонованого SaaS та відстеження реклами, а також автоматичне втручання. Дізнайтеся, що таке управління ШІ .

4. Виявлення шпигунського та шкідливого програмного забезпечення за допомогою карантину: перехоплює трафік керування та контролю, щоб зупинити витік, допомагає користувачам видалити його та автоматично поміщає пристрої, що знаходяться в зоні ризику, у карантин відповідно до специфікацій вашої політики.

5. Довіра до пристроїв та умовний доступ: оцінює стан як корпоративних, так і особистих пристроїв і блокує доступ з будь-якого пристрою, який виходить з ладу.

6. Розгортання: безконтактне керування для керованих та некерованих пристроїв, з ручною реєстрацією електронною поштою або SMS для парків пристроїв BYOD та організацій без MDM.

7. Інтеграції: Microsoft Intune та VMware Workspace ONE для управління,

Microsoft, Google та Okta для ідентифікації, а також Microsoft Defender для кінцевих точок для XDR. Сертифіковано за стандартом ISO 27001.

8. Архітектура, що зберігає конфіденційність: немає відстеження місцезнаходження, немає сканування файлів, немає історії переглядів та немає доступу до вмісту повідомлень.

Check Point Harmony Mobile

Check Point Harmony Mobile – це компонент захисту від мобільних загроз пакету Check Point Harmony. Він захищає пристрої iOS та Android від атак програм, мережі та ОС, а також інтегрується з існуючими платформами UEM для забезпечення відповідності вимогам та забезпечення дотримання політик.



Рисунок 2.2 – Інтерфейс користувача Check Point Harmony Mobile

Він запобігає зараженню пристроїв співробітників шкідливим програмним забезпеченням, виявляючи та блокуючи завантаження шкідливих програм у режимі реального часу. Він блокує фішинг, що поширюється через

будь-яку програму, запобігає атакам типу «посередник» на стільникові мережі та мережі Wi-Fi, блокує доступ заражених пристроїв до корпоративних даних, а також розпізнає та блокує передові методи джейлбрейка.

CrowdStrike Falcon для мобільних пристроїв

CrowdStrike Falcon для мобільних пристроїв розширює можливості виявлення та реагування кінцевих точок платформи Falcon на iOS та Android, керуючи з тієї ж консолі, що й робочі станції та сервери. CrowdStrike позиціонує його як мобільний EDR, а не як звичайний MTD.

Він виявляє підозрілу активність та посилання в текстових повідомленнях, електронній пошті, браузерах та QR-кодах, виявляє мобільне шкідливе програмне забезпечення, збої в мережі, підроблені особисті дані та пристрої з джейлбрейком, а також може містити уражені пристрої з консолі.

Він пропонує вбудований список блокування шкідливих IP-адрес для зменшення ризику, блокує шкідливі посилання та неавторизовані домени, а також виявляє мобільне шкідливе програмне забезпечення, збої в мережі, підроблені ідентифікаційні дані, пристрої з джейлбрейком та випадкове витікання даних.

Ivanti Neurons для захисту від мобільних загроз

Ivanti Neurons for MTD захищає пристрої Android, iOS та iPadOS, як корпоративні, так і ті, що належать співробітникам. Можливості Ivanti MTD побудовані на технології Zimperium, а для налаштування потрібна консоль керування Zimperium. Виявлення та виправлення виконуються як на пристрої, так і в хмарі.

Він контролює параметри та конфігурацію пристрою, мережевий трафік та встановлені програми, запобігає фішингу за допомогою аналізу з’єднань у режимі реального часу та застосовує локальні дії щодо відповідності вимогам для усунення загроз на пристрої.

Jamf Protect для мобільних пристроїв

Jamf – це компанія з управління пристроями Apple, яка додала захист від мобільних загроз завдяки придбанню Wandera у 2021 році та перебуває у

приватній власності Francisco Partners з січня 2026 року. Jamf Protect охоплює macOS, iOS, iPadOS та Android в одному продукті, при цьому перевірка трафіку здійснюється через безпечний DNS за замовчуванням, а для глибшого контролю доступності використовується опція на основі проксі-сервера.

Він блокує фішинг, криптоджекінг та шкідливі домени в режимі реального часу, позначає програми з небезпечними дозволами або підозрілими профілями розробників, фільтрує веб-контент на відповідність прийнятним політикам використання, повідомляє про використання стільникових даних програмами, оцінює вразливості ОС та програм на основі даних про серйозність CVE та передає ризики пристроїв в умовний доступ.

Jamf Mobile Forensics – це окреме доповнення, що охоплює шпигунські програми-найманці та атаки з нульовим кліком для користувачів з високим рівнем ризику.

Безпека мобільних кінцевих точок Lookout

Lookout Mobile Endpoint Security захищає пристрої iOS, Android та ChromeOS, а також виявляє фішинг в електронній пошті, SMS, додатках для обміну повідомленнями та соціальних мережах.

Він оцінює програми на наявність шкідливого програмного забезпечення, ризиків для конфіденційності та безпеки, виявляє вразливі операційні системи, шифрує та захищає DNS-трафік, виявляє мережеві атаки та складні шпигунські програми, а також враховує стан пристрою в рішеннях щодо умовного доступу.

Він виявляє рутування, джейлбрейк, втручання в ОС та передові методи експлуатації. Він також виявляє шахрайські мережі Wi-Fi, атаки типу «людина посередині» та небезпечні мережеві умови в режимі реального часу.

Microsoft Defender для кінцевих точок

Microsoft Defender для кінцевих точок забезпечує захист від веб-сторінок та фішингу, а також блокує небезпечні мережеві підключення на Android та iOS. Виявлення шкідливого програмного забезпечення доступне на Android, а виявлення джейлбрейка – на iOS, а Microsoft Defender Vulnerability Management оцінює обидва типи програм.

Ризик для пристроїв залежить від політик відповідності Intune та умовного доступу, а на iOS Microsoft документує варіант розгортання, який використовує захист програм, а не реєстрацію пристроїв.

Захист від мобільних загроз Pradeo Security

Pradeo – французька компанія з розробки мобільної безпеки, чия платформа MTD захищає Android, iOS та Chromebook. Вона виявляє загрози, що надходять від програм, мережі та ОС, включаючи шкідливе програмне забезпечення, шпигунське програмне забезпечення, нав'язливі програми та фішинг, та усуває їх без втручання адміністратора.

Pradeo серед своїх можливостей перераховує захист мережі від атак типу «людина посередині», шкідливих ретрансляційних антен та шкідливих проксі-з'єднань, а також виявлення застарілих версій операційних систем та пристроїв з root-доступом або джейлбрейком.

Мобільна версія SentinelOne Singularity

Singularity Mobile – це модуль захисту від мобільних загроз SentinelOne, який охоплює iOS, Android та ChromeOS з тієї ж консолі, що й продукти для кінцевих пристроїв. Він описує своє виявлення як на пристрої та на основі штучного інтелекту, і стверджує, що продукт працює з MDM або без нього. SentinelOne під час запуску оголосила про продукт на базі Zimperium, і Zimperium перераховує SentinelOne серед своїх партнерів.

Він виявляє шкідливе програмне забезпечення нульового дня, фішинг та атаки типу «людина посередині», постійно перевіряє програми на наявність ризиків для конфіденційності та безпеки, а також блокує мережеві атаки (включаючи зловмисні Wi-Fi) та небезпечні комунікації.

Zimperium Mobile Threat Defense

Zimperium Mobile Threat Defense захищає пристрої, що належать компанії, та пристрої, що виготовляються самостійно, на iOS, Android та ChromeOS. Компанія стверджує, що її технологія машинного навчання на пристрої виявляє відомі загрози та загрози нульового дня в режимі реального часу та продовжує працювати, коли пристрій не має підключення до мережі. Zimperium також

ліцензує свою технологію виявлення іншим постачальникам, зокрема Ivanti.

Він виявляє фішинг, спрямований на мобільні пристрої, оцінює ризики як корпоративних, так і персональних додатків, виявляє шахрайські та небезпечні мережі, перевіряє цілісність пристроїв та підтримує судово-медичне сканування уражених пристроїв.

Вибір програмного забезпечення для захисту від мобільних загроз

Продукти на цьому ринку більше схожі за своїм маркетингом, ніж за дизайном. Ці питання, як правило, розділяють їх:

– Що вже покриває ваша існуюча платформа? Якщо ви вже ліцензували пакет керуючих або кінцевих точок, який включає виявлення мобільних пристроїв, питання не в тому, який продукт є найсильнішим окремо, а в тому, чи виправдовує додаткове покриття другий контракт і другу консоль.

– Чи входять до сфери дії некеровані пристрої? Підрядники, члени правління, сезонний персонал та працівники, які відмовляються реєструвати особистий телефон, отримують доступ до даних компанії на пристроях, якими ви не керуєте. Деякі продукти добре справляються з цим, а деякі передбачають реєстрацію. Якщо ця група користувачів важлива для вас, протестуйте її спеціально, а не довіряйте заяві на основі даних.

– Де виконується виявлення? Аналіз на пристрої продовжує працювати навіть за втрат зв’язку та зберігає локальний трафік. Хмарний аналіз дозволяє виконувати інтенсивнішу обробку та кореляцію між усіма пристроями. Жоден з варіантів не є автоматично кращим, але вибір впливає на покриття офлайн, використання заряду батареї та даних, а також на те, що залишає пристрій.

– Що система збирає з особистих телефонів? Це питання визначає, чи буде розгортання успішним. Працівники не погодяться на інструмент, який, на їхню думку, читає їхні повідомлення або відстежує їхній перегляд веб-сторінок. Запитайте, яка телеметрія збирається, що організація може бачити в консолі, а що залишається для неї невидимим.

– Вам потрібне розслідування чи лише виявлення? Якщо телефон скомпрометовано, що потрібно робити вашій команді далі? Деяким організаціям

потрібне сповіщення та блокування. Іншим потрібно встановити, до чого було здійснено доступ, протягом якого періоду, та надати докази регулятору чи страховику. Ця відмінність вказує на різні продукти.

– Де обробляються та зберігаються дані? Сектор та юрисдикція можуть обмежувати це. Встановіть, які регіони обробляють телеметрію, хто є субпідрядниками, і чи є відповідь договірною чи просто поточною практикою.

– Як це поєднується з тим, що ви вже використовуєте? Перевірте інтеграції, які ви фактично використовуватимете: вашу платформу управління, ваші політики умовного доступу та ваш SIEM або конвеєр реєстрації. Запитайте, чи це підтримувані продукти, чи проект професійних послуг.

– Чи побудовано це на движку іншого постачальника? Деякі продукти на цьому ринку ліцензують технологію виявлення від інших постачальників. Саме по собі це не є проблемою, але варто це знати, оскільки організація, яка використовує продукт управління однією платформою та продукт для виявлення загроз іншої, може зрештою двічі придбати один і той самий движок виявлення. Запитайте безпосередньо, чия технологія виконує виявлення.

– Як ви дізнаєтесь, що це працює? Узгодьте докази, перш ніж підписувати. Запустіть пілотний проект на реальних пристроях у вашому власному середовищі, дотримуючись власних правил. Запитайте, чого продукт не вистачає, і остерігайтеся будь-якої відповіді, яка натякає, що він нічого не вистачає.

Висновок

Тепер телефони містять електронну пошту, схвалення, дані клієнтів та інструменти штучного інтелекту, які потрібні зловмисникам, а багато елементів керування, що захищають ноутбуки та електронну пошту, ніколи не бачать, що на них надходить. Керування пристроями підтримує конфігурацію телефонів та їх відповідність вимогам, а захист від мобільних загроз додає виявлення та блокування, які управління ніколи не було розроблено для забезпечення.

Постачальники послуг у цьому посібнику найбільше відрізняються тим, де виконується виявлення, що вони збирають з персональних пристроїв і

наскільки далеко вони виходять за рамки сповіщення. Кілька з них мають спільний базовий механізм виявлення, тому варто запитати, чия технологія виконує виявлення, перш ніж порівнювати функції.

Правильний вибір залежить від того, що вже включають ваші існуючі ліцензії, чи входять до їх дії некеровані та особисті пристрої, а також які докази потрібні вашій організації після інциденту.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python є ефективним середовищем розробки, яке поєднує високорівневі структури даних із гнучкими об'єктно-орієнтованими парадигмами, що є критично важливим для швидкого прототипування та моделювання алгоритмів криптографічного захисту й скремблювання даних. Завдяки динамічній типізації та виразному синтаксису мова оптимально підходить для реалізації логіки обробки цифрових сигналів на кросплатформних апаратних архітектурах.

Офіційні дистрибутиви та розширювані компоненти інтерпретатора Python доступні на веб-сайті, що забезпечує інтеграцію необхідних бібліотек для розгортання та тестування програмних модулів. Завдяки відкритій архітектурі інтерпретатор легко доповнюється низькорівневими функціями та типами даних, написаними на C або C++, що дозволяє оптимізувати обчислювально складні операції з бітовими послідовностями та реалізовувати гібридні програмно-апаратні механізми скремблювання в середовищі мобільних операційних систем.

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 програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

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

Операційні системи (ОС) для мобільних пристроїв слугують основою смартфонів і планшетів, дозволяючи їм виконувати численні завдання та запускати програми. З іншого боку, швидке розширення мобільних технологій створює кілька перешкод у розробці та розгортанні цих операційних систем. Розуміння та вирішення цих проблем має вирішальне значення як для розробників, виробників, так і для користувачів. Давайте глибше заглибимося у складність ландшафту операційних систем мобільних пристроїв.

Тонкощі проектування та впровадження операційних систем мобільних пристроїв

1. Обмеженість ресурсів: балансування

Мобільні пристрої працюють в умовах жорстких обмежень ресурсів, таких як обмежена пам'ять, обчислювальна потужність та час роботи від акумулятора. Завдання розробників ОС полягає в оптимізації використання ресурсів без шкоди для продуктивності чи функціональності. Пошук тонкого балансу між ефективністю та багатим набором функцій є постійною боротьбою.

2. Навігація фрагментацією пристроїв: забезпечення сумісності

Фрагментація пристроїв виникає через різноманітні апаратні конфігурації на різних пристроях. Розробка ОС, яка враховує цю різноманітність, зберігаючи при цьому узгодженість продуктивності, зручності використання та сумісності програм, створює значні перешкоди. Проблеми сумісності, що виникають через різні розміри екранів, процесори та можливості апаратного забезпечення, потребують ефективного вирішення.

3. Зміцнення безпеки: боротьба з кіберзагрозами

Мобільні пристрої містять скарбницю конфіденційних даних, що робить їх головними цілями для кіберзагроз. Розробники ОС стикаються з проблемою

впровадження надійних заходів безпеки для захисту від шкідливого програмного забезпечення, витоків даних та вразливостей без шкоди для продуктивності пристрою чи зручності користувача.

4. Конфіденційність користувачів у фокусі: досягнення правильного балансу

Збір даних користувачів для персоналізованого досвіду має бути збалансований з урахуванням питань конфіденційності користувачів. Розробники ОС повинні створювати функції, які надають персоналізовані послуги, водночас забезпечуючи прозорість та контроль користувачів над їхніми даними, вирішуючи поточну дискусію між персоналізацією та конфіденційністю.

5. Безперебійні оновлення операційної системи: проблема координації

Своєчасне та безперебійне оновлення для різних моделей пристроїв та від різних виробників є складним завданням. Координація оновлень, забезпечення сумісності, усунення потенційних помилок або збоїв і мінімізація збоїв у взаємодії з користувачем залишаються значними викликами.

6. Оптимізація багатозадачності: підтримка стабільності

Підтримка багатозадачності та забезпечення стабільності системи є багатогранним завданням. Досягнення плавного переходу між програмами та ефективного розподілу ресурсів для одночасних завдань без шкоди для продуктивності або розрядки батареї вимагає постійного вдосконалення.

7. Гармонія UI/UX: на різних пристроях

Створення інтуїтивно зрозумілого інтерфейсу користувача (UI) та забезпечення виняткового користувацького досвіду (UX) на широкому спектрі пристроїв є постійним викликом. Адаптація до різних розмірів екранів, роздільної здатності та форм-факторів, забезпечуючи при цьому узгоджений та зручний інтерфейс, вимагає постійних інновацій.

Вирішення проблем: стратегічні підходи та інновації

1. Інновації в управлінні ресурсами

Ефективне управління ресурсами за допомогою інтелектуальних алгоритмів та оптимізацій залишається критично важливим. Такі методи, як визначення пріоритетів завдань, стиснення пам'яті та протоколи енергозбереження, є важливими для максимізації ефективності пристроїв.

2. Уніфіковані платформи розробки для забезпечення сумісності

Розробка уніфікованих платформ розробки, що сприяють сумісності між пристроями, може зменшити проблеми фрагментації. Заохочення стандартизованої розробки додатків на різних пристроях спрощує проблеми сумісності.

3. Посилені протоколи безпеки

Впровадження надійного шифрування, протоколів безпечного завантаження та своєчасних виправлень безпеки підвищує безпеку пристроїв. Розширені механізми автентифікації та методи «пісочниці» пропонують рівні захисту від потенційних загроз.

4. Протоколи конфіденційності, орієнтовані на користувача

Впровадження прозорих практик збору даних та налаштувань конфіденційності, контрольованих користувачем, гарантує користувачам конфіденційність їхніх даних. Інтеграція методів анонімізації та детальних механізмів контролю даних посилює заходи конфіденційності.

5. Оптимізовані стратегії оновлення

Співпраця з виробниками пристроїв та операторами зв'язку має вирішальне значення для оптимізації процесів оновлення. Оновлення через бездротову мережу (OTA) та поетапне розгортання забезпечують своєчасну доставку без шкоди для сумісності пристроїв.

6. Інноваційні практики UI/UX

Використання принципів адаптивного дизайну та адаптивних компонентів UI/UX забезпечує цілісний досвід. Ітерації дизайну, орієнтовані на користувача, та тестування зручності використання вдосконалюють та

покращують загальний інтерфейс.

Розробка та розгортання операційних систем мобільних пристроїв – це складне завдання, сповнене труднощів. Постійні інновації та адаптивність необхідні через поєднання обмежень ресурсів, фрагментації пристроїв, проблем безпеки, конфіденційності користувачів, безперебійної доставки оновлень, оптимізації багатозадачності та надзвичайного користувацького досвіду.

Постійно мінливий світ мобільних технологій завжди ставить нові виклики. Однак вирішення цих проблем виводить мобільні операційні системи вперед завдяки проривам у методах розробки, технологічних розробках та спільних ініціативах. Мобільні операційні системи продовжують розвиватися, впроваджуючи новітні рішення та вміло долаючи труднощі, надаючи споживачам складні функції та безперебійний досвід роботи на різних пристроях.

3.2 Розробка структурної схеми

Більшість команд безпеки витратили десятиліття на створення захисту навколо ноутбуків та серверів. Тим часом робота перемістилася на телефони. Тепер співробітники читають електронні листи, схвалюють платежі, відкривають записи клієнтів та приєднуються до зустрічей на пристроях, які знаходяться поза межами майже всіх контролів, що належать команді безпеки.

Програмне забезпечення для захисту від мобільних загроз існує для того, щоб заповнити цю прогалину. Воно входить до ширшої галузі програмного забезпечення для мобільної безпеки та є окремим від управління пристроями. Ця різниця важливіша, ніж випливає з назви.

Що таке мобільна безпека та навіщо вона потрібна

Мобільна безпека – це практика захисту телефонів і планшетів, а також корпоративних даних, до яких вони отримують доступ, від атак та втрати даних.

Це не просто безпека робочого столу на меншому екрані, оскільки мобільні операційні системи побудовані по-різному. Програми працюють

усередині пісочниць, і жодна з платформ не дозволяє доступ на рівні ядра, на який покладаються інструменти безпеки ноутбуків. Натомість вони пропонують набір вузьких, санкціонованих інтерфейсів, на яких побудовані продукти мобільної безпеки.

Ці дві платформи відрізняються обсягом дозволеного сигналу. Android дозволяє перевірку встановлених програм, тому сканування програм там можливе. iOS цього не дозволяє, тому покриття на пристроях Apple більше залежить від мережевих сигналів та конфігурації пристрою.

Вразливість також різна. Телефон постійно залишає корпоративну мережу. Він підключається до Wi-Fi готелів та аеропортів, переміщується між мобільними мережами та отримує повідомлення через канали, які не перевіряє жоден шлюз електронної пошти. Деякі телефони, що надходять до корпоративних даних, належать особисто, що обмежує можливості організації щодо їхнього контролю. Результатом є поверхня для атаки, яка інтенсивно використовується, слабо контролюється та доступна будь-кому.

Фішинг – це найбільша загроза, і на мобільних пристроях вона потрапляє через набагато більше дверей, ніж електронна пошта. Посилання надходять у текстових повідомленнях, WhatsApp, LinkedIn, Signal, додатках для знайомств та голосових дзвінках. Маленькі екрани ускладнюють їх виявлення. Адреси обрізаються, немає попереднього перегляду при наведенні курсора, а мобільні інтерфейси розроблені так, щоб приховувати деталі, а не показувати їх.

QR-коди йдуть ще далі. Код на плакаті, рахунку-фактурі чи паркоматі повністю знімає атаку з корпоративної мережі, оскільки людина сканує його за допомогою персонального застосунку для камери та потрапляє на сторінку, яка не перевіряється. Це називається квишингом .

Додатки – це ще один шлях. Огляди в магазинах додатків охоплюють багато чого, хоча й не все, а завантаження збоку повністю обходить огляди. Легітимні додатки створюють власні ризики, коли вони запитують більше доступу, ніж їм потрібно, або надсилають дані в місця, які організація не санкціонує.

Мережі – це ще один спосіб. Шахрайські точки доступу, атаки зниження версії протоколу та маніпуляції сертифікатами дозволяють зловмиснику читати або змінювати трафік, який користувач вважає захищеним.

Також є пристрої без патчів. Мобільні операційні системи часто отримують оновлення безпеки, але користувачі їх відкладають, а старіші телефони Android взагалі перестають їх отримувати, що залишає відомі вразливості відкритими на пристроях, які все ще щодня отримують доступ до систем компанії.

Комерційне шпигунське програмне забезпечення знаходиться на найважчому кінці списку. Інструменти спостереження, що продаються урядовим клієнтам, використовуються проти журналістів, активістів, чиновників та керівників вищої ланки. Деякі з них взагалі не потребують взаємодії з боку цілі.

Зрештою, тепер співробітники вставляють інформацію про компанію в споживчі додатки зі штучним інтелектом та інтерфейси чату на своїх телефонах. Дані залишаються без будь-якого контролю, який належить організації.

Програмне забезпечення для мобільної безпеки охоплює широкий спектр інструментів для вирішення цих ризиків, від споживчих антивірусних програм до корпоративних платформ. Деякі з цих платформ керують пристроями. Категорія, створена для виявлення та блокування вищезазначених загроз, – це захист від мобільних загроз.

Що таке програмне забезпечення для захисту від мобільних загроз?

Захист від мобільних загроз, скорочено MTD, – це програмне забезпечення, яке виявляє та блокує загрози на пристроях iOS та Android. Зазвичай воно працює на трьох рівнях:

– На рівні пристрою він перевіряє стан самого телефону, шукаючи джейлбрейк або рутування, відсутні оновлення безпеки, слабку конфігурацію та ознаки компрометації.

– На рівні додатків він оцінює встановлені додатки на наявність шкідливої поведінки, надмірних дозволів та небезпечної обробки даних.

– На мережевому рівні він перевіряє з'єднання, щоб виявляти спроби

перехоплення та блокувати трафік до фішингової та командно-контрольної інфраструктури.

Розгортання зазвичай відбувається через додаток, що розгортається будь-якою платформою управління, яку вже використовує організація, хоча більшість продуктів також підтримують пристрої, які взагалі не зареєстровані в системі управління. Це важливо для підрядників та для персоналу, який відмовляється реєструвати особистий телефон.

Місце проведення аналізу залежить від продукту, і це одна з найважливіших відмінностей між ними. Деякі виконують виявлення на самому пристрої, тому рішення про загрозу приймається локально, а не в хмарі, що означає, що захист продовжується, навіть коли телефон не має підключення до мережі, а оцінюваний контент ніколи не надсилається на аналіз. Інші надсилають телеметрію до хмарного сервісу, який централізовано оцінює ризики, що дозволяє проводити ретельніший аналіз та кореляцію по всьому парку пристроїв. Деякі поєднують ці два методи.

Більшість продуктів також враховують певний сигнал у рішеннях щодо доступу. Якщо пристрій вважається ризикованим, політики умовного доступу можуть блокувати його доступ до електронної пошти, файлів або бізнес-програм, доки проблему не буде вирішено.

Варто чітко розуміти обмеження. Захист від мобільних загроз не керує пристроями, не розповсюджує програмне забезпечення, не забезпечує дотримання конфігурації, а також не замінює засоби контролю безпеки, що використовуються для керування вашими програмами та даними. Він виявляє загрози, що досягають телефону, та реагує на них.

MTD, MDM, UEM та мобільний EDR: у чому відмінності

Ці чотири терміни використовуються нечітко, і їх збіг може призвести до справжньої плутанини під час закупівель:

Керування мобільними пристроями, або MDM, займається реєстрацією, налаштуванням та політиками. Воно встановлює правила використання паролів, надсилає програми та сертифікати, відокремлює робочі дані від особистих та

стирає дані з пристрою, коли хтось його залишає. MDM адмініструє телефон, але не створено для виявлення атаки на нього.

Уніфіковане керування кінцевими точками, або UEM, – це ширша версія тієї ж ідеї. Воно об'єднує телефони, планшети, ноутбуки та настільні комп'ютери в одній консолі. Кілька продуктів UEM поєднують у собі деякі функції виявлення загроз, яких часто достатньо для парків пристроїв з низьким рівнем ризику, і варто з'ясувати, що вже включено до вашого продукту, перш ніж купувати щось додаткове.

Захист від мобільних загроз (MTD) працює поряд з обома. Він може припускати, що пристроями керують, і зосереджується на виявленні загроз та їх зупинці. У більшості розгортань MTD та MDM працюють разом та обмінюються інформацією, а платформа управління діє на основі того, що виявляє продукт із загрозами.

Виявлення та реагування на мобільні кінцеві точки, або мобільне EDR, – це новіша концепція, запозичена зі світу ноутбуків. Акцент зміщується з виявлення на розслідування: збір телеметрії операційної системи, реконструкція того, що сталося на пристрої, та підтримка судово-медичного реагування. Кілька відомих постачальників зараз використовують цю назву для продуктів, які значно відрізняються тим, що вони фактично збирають та підтримують, тому вона говорить вам менше, ніж базова функція.

Ще одну категорію часто плутають із цією. Захист у додатку, який іноді називають екрануванням додатків або безпекою мобільних додатків, захищає програмне забезпечення, яке ваша організація публікує для своїх клієнтів. Він зміцнює ваш додаток від несанкціонованого доступу та зворотного проектування.

Аналіз можливих проблем

1. Інформаційна безпека та захист конфіденційності:

– Ризик витоку конфіденційної інформації з зустрічей: мікрофони смартфонів можуть бути активовані шкідливим програмним забезпеченням для запису конфіденційного контенту зустрічі.

– Випадкова активація голосового помічника: Siri, Google Assistant та інші помічники зі штучним інтелектом можуть випадково активуватися під час розмов, записуючи приватні обговорення.

– Захист від конкурентної розвідки: Організаціям необхідно запобігати захопленню бізнес-секретів через мікрофони смартфонів.

2. Вимоги до дотримання нормативних вимог:

– Суворі вимоги до зони безпеки: деякі урядові, військові або науково-дослідні установи забороняють аудіозапис, але водночас дозволяють спілкування.

– Конфлікти політики BYOD: організації дозволяють співробітникам приносити власні пристрої, але повинні контролювати ризики запису.

– Специфічні для галузі вимоги до дотримання вимог: юридичний, медичний та фінансовий сектори повинні знаходити баланс між потребами в комунікації та захистом конфіденційності клієнтів.

Ці сегменти клієнтів надають великого значення захисту інформаційної безпеки, водночас стикаючись із суворими вимогами дотримання нормативних вимог, що створює чіткий попит на безпечні телефонні блоки, які блокують можливості аудіозапису, зберігаючи при цьому підключення до мережі.

Обмеження традиційних рішень для запобігання прослуховуванню

Традиційні підходи до безпеки смартфонів під час конфіденційних зустрічей мають суттєві недоліки:

1. Повне вимкнення пристрою: Вимкнення телефонів повністю усуває ризики запису, але створює кілька проблем:

– Учасники пропускають термінові дзвінки та повідомлення.

– Екстрені сповіщення не надходять.

– Перезапуск пристроїв займає дорогоцінний час.

– Деякі шкідливі програми можуть активувати пристрої, навіть якщо вони нібито вимкнені.

2. Режим польоту: Хоча це краще, ніж вимкнення, це рішення:

– Блокує всі комунікації, включаючи критичні повідомлення.

– Вимагає від користувачів пам'ятати про активацію/деактивацію.

– Все ще дозволяє деякі можливості запису голосу.

– Не запобігає офлайн-запису зі скомпрометованих пристроїв.

3. Політика збору: Здача пристроїв учасниками:

– Створює адміністративне навантаження.

– Викликає занепокоєння щодо безпеки пристроїв.

– Викликає тривогу через пропущені повідомлення.

– Часто зустрічає опір з боку керівників та VIP-персон.

4. Сумки Фарадея: Блокують всі сигнали:

– Виключіть корисне мережеве підключення.

– Часто мають непослідовну ефективність.

– Може бути громіздким у використанні.

– Повністю запобігти екстреним зв'язкам.

У даній роботі для захисті мобільних телефонів використовується алгоритм AES.

Структурна схема наведена на рисунку 3.1. Структурна схема системи представлена на рисунку 3.1. Як видно зі схеми, розроблене рішення складається з таких основних компонентів:

1. Дані, що передаються між мобільними пристроями.

2. Блок шифрування на базі алгоритму AES для скремблювання цифрового сигналу на мобільному пристрої.

3. Блок розшифрування на базі алгоритму AES для зворотної обробки цифрового сигналу.

4. Мобільний пристрій-одержувач, на який надходять зашифровані дані.

Ключовим елементом системи є блок шифрування AES, який варто розглянути детальніше.

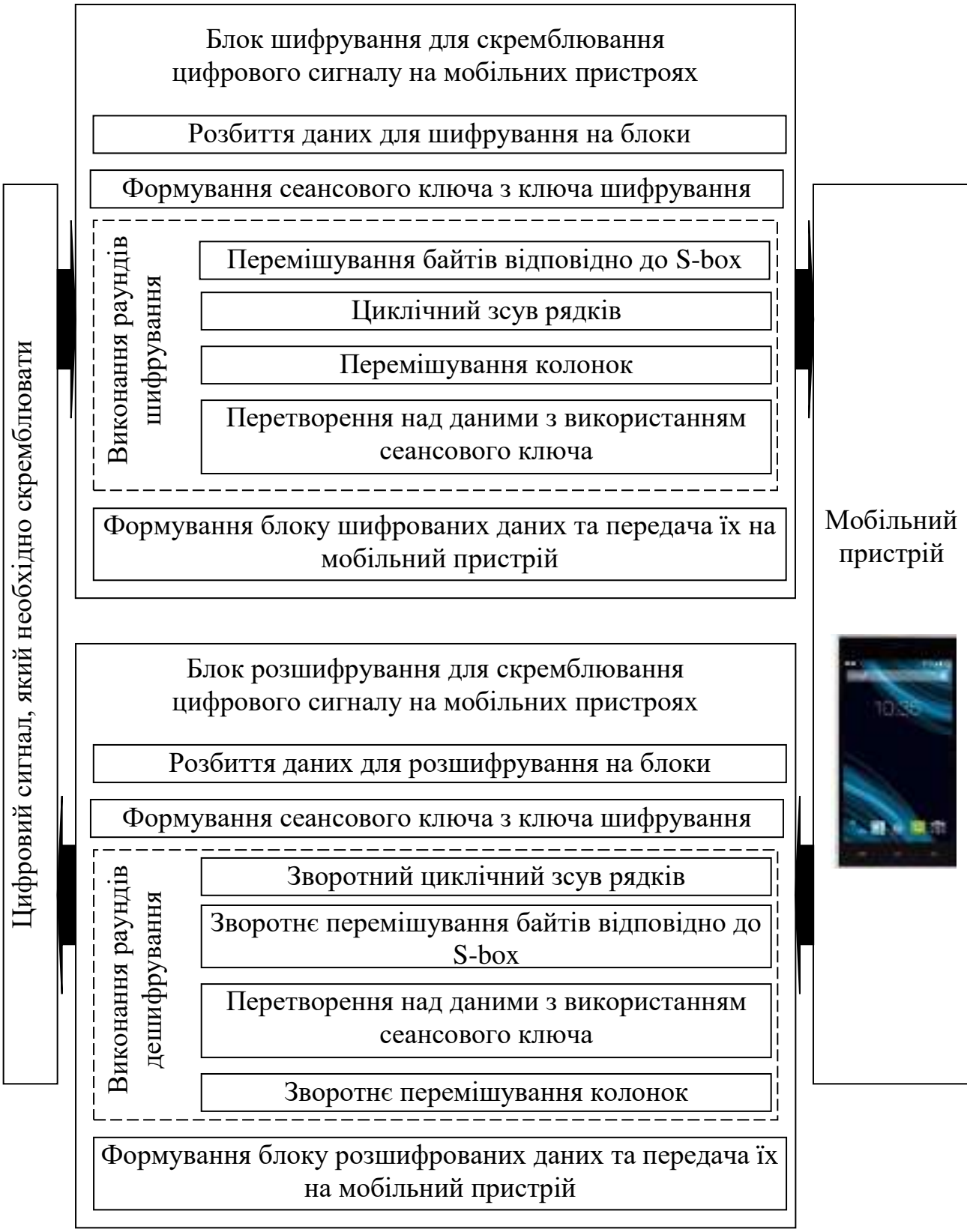

Рисунок 3.1 – Структурна схема системи

Алгоритм шифрування AES виконується за таким алгоритмом:

1. Вхідні дані (input) розбиваються на блоки та згідно із заданим правилом копіюються до робочого масиву State.

2. За допомогою функції KeyExpansion() із початкового ключа (Cipher Key) генерується набір раундових ключів (Round Key).

3. Визначається кількість раундів (10, 12 або 14), що залежить від довжини використаного ключа.

4. Проводиться ітераційне шифрування протягом визначеної в п. 3 кількості раундів:

– SubBytes() – нелінійна заміна байтів масиву State за допомогою таблиці підстановки (S-box), яка застосовується до кожного байта незалежно;

– ShiftRows() – циклічний зсув останніх трьох рядків масиву State на визначену кількість позицій;

– MixColumns() – перемішування даних у стовпцях масиву State для забезпечення дифузії;

– AddRoundKey() – побітове додавання за модулем 2 (XOR) поточного раундового ключа (Round Key) до масиву State (довжина ключа дорівнює розміру State, наприклад, 128 біт або 16 байт для Nb = 4).

5. Після завершення фінального раунду трансформації вміст масиву State за встановленим правилом копіюється у вихідний блок (output).

Алгоритм розшифрування AES реалізується у зворотній послідовності:

1. Зашифровані вхідні дані (input) розбиваються на блоки та за відповідним правилом копіюються до масиву State.

2. За допомогою функції KeyExpansion() з ключа шифрування (Cipher Key) повторно формуються раундові ключі (Round Key).

3. Встановлюється кількість раундів (10, 12 або 14) відповідно до довжини ключа.

4. Виконується операція розшифрування шляхом проходження раундів у зворотному порядку:

– InvShiftRows() – зворотна циклічна перестановка рядків масиву State;

– InvSubBytes() – зворотна нелінійна заміна байтів із використанням оберненої таблиці S-box;

– InvAddRoundKey() – зворотне раундове додавання ключа (XOR) до масиву State;

– InvMixColumns() – зворотне перемішування стовпців масиву State.

5. Після завершення останнього раунду зворотних трансформацій масив State копіюється у вихідний блок розшифрованих даних (output).

3.3 Розробка функціональної схеми

На рисунку 3.2 наведено функціональну схему розробленої системи. Архітектура програмного забезпечення охоплює такі взаємопов'язані модулі:

1. Головний інтерфейс користувача – забезпечує навігацію та доступ до всіх функціональних підсистем комплексу.

2. Модуль логічного розділення пам'яті мобільного пристрою – здійснює сегментацію сховища на відкриту (незашифровану) та захищену області.

3. Модуль автентифікації та перевірки легітимності ідентифікатора (паролю) – відповідає за зчитування, хешування та верифікацію ключової інформації з можливістю її безпечної зміни.

4. Модуль моніторингу та лічильника невдалих спроб автентифікації – реалізує захист від брутфорс-атак шляхом обмеження кількості ітерацій введення пароля.

5. Модуль симетричного шифрования (AES) – виконує потокове скремблювання та шифрування цифрових сигналів і файлових масивів у середовищі мобільної ОС.

6. Модуль дешифрування (AES) – забезпечує зворотну процедуру розшифрування та відновлення цілісності сигналу для легітимного користувача.

7. Модуль гарантованого знищення конфіденційних даних – реалізує алгоритм Гутмана для безповоротного стирання інформації при спрацьовуванні захисних тригерів.

8. Довідково-інформаційний модуль – містить контекстну довідку, метадані про розробника, версію та дату релізу програмного продукту.

9. Головне вікно програми

Головний інтерфейс є єдиною точкою управління захищеним середовищем та містить:

– назву програмного модуля;

– рядок головного меню;

– панель інструментів швидкого доступу;

– робочу область;

– рядок стану (Status Bar) для відображення поточних режимів шифрування/скремблювання.

10. Блок розбиття мобільного пристрою на незашифровану та зашифровану частини. Забезпечує створення захищеного контейнера на носії даних. На відкритій частині розміщується мінімальний сервісний додаток для ініціалізації доступу до зашифрованого сегмента. Обидві частини монтуються під єдиним ім'ям із динамічним перемиканням, що виключає паралельний несанкціонований доступ до захищеного контенту.

11. Блок зчитування та перевірки на легітимність паролю. Здійснює верифікацію введеного користувачем пароля шляхом порівняння з еталонним значенням. Передбачено механізм зміни парольної фрази, який вимагає обов'язкового введення чинного пароля та підтвердження нового.

12. Блок підрахунку спроб введення некоректного паролю. Реалізує політику безпеки мобільного додатка: після 10 послідовних невдалих спроб введення ключа доступу ініціюється примусове блокування системи та активація процедури знищення даних.

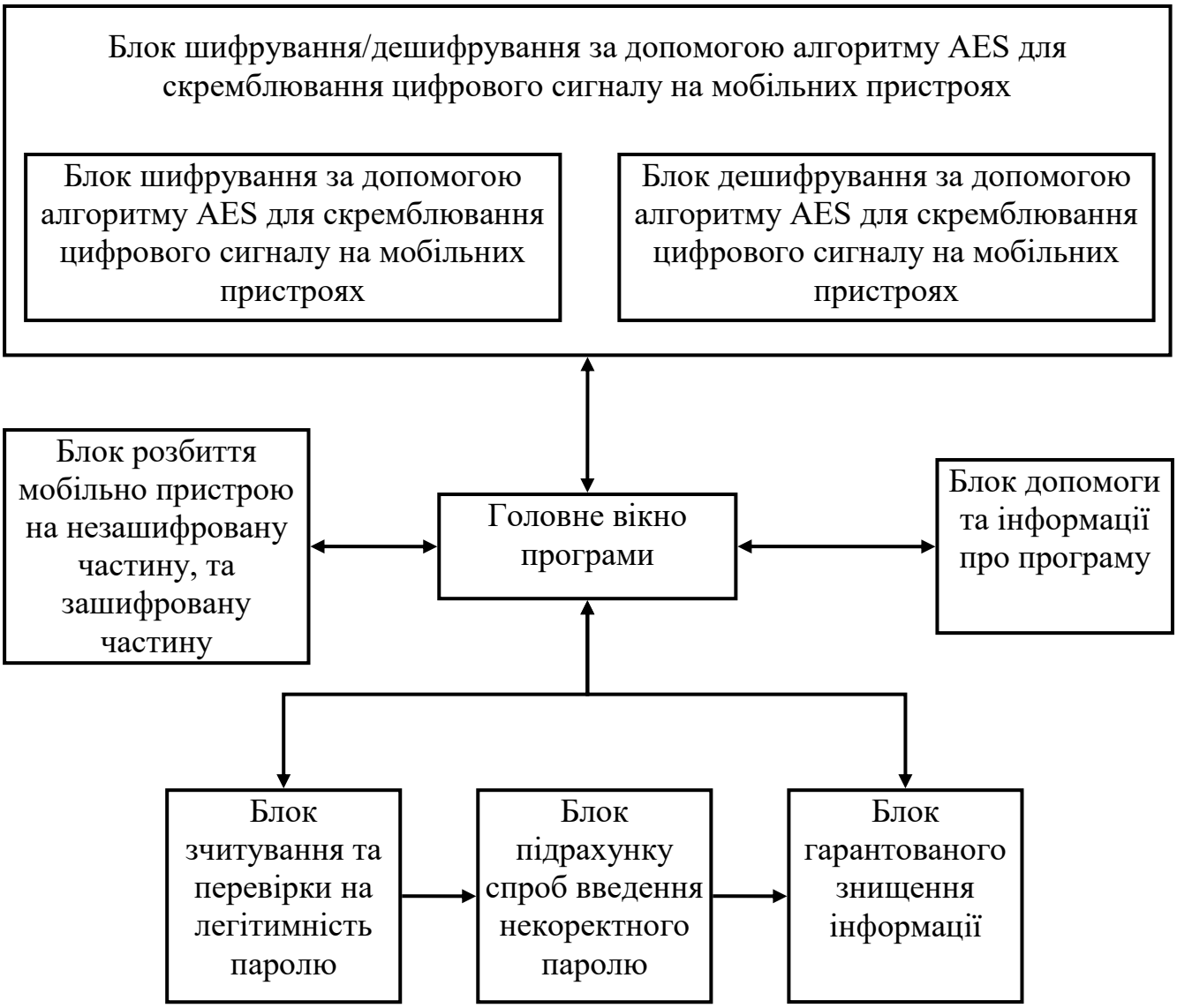

Рисунок 3.2 – Функціональна схема програмного забезпечення

13. Блоки шифрування та дешифрування за допомогою алгоритму AES. Здійснюють криптографічне перетворення (скремблювання) цифрових сигналів та бітових послідовностей на базі криптоалгоритму AES (Advanced Encryption Standard), забезпечуючи високу стійкість до перехоплення та мінімальні накладні витрати на обчислення в мобільній ОС.

14. Блок гарантованого знищення інформації. Призначений для безповоротного стирання даних у разі критичних інцидентів безпеки (перевищення ліміту спроб введення пароля). В основі реалізації лежить модифікований алгоритм Гутмана, який передбачає багатократний перезапуск

циклів запису псевдовипадкових послідовностей, нульових байтів (#00) та бітових масок (#FF) на фізичні сектори носія. Це нівелює ризики відновлення інформації навіть із використанням спеціалізованих апаратних засобів аналізу залишкової намагніченості.

15. Блок допомоги та інформації про програму. Містить довідкову документацію щодо експлуатації комплексу, а також службові відомості про розробника, номер збірки та дату випуску програмного забезпечення.

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 потоків даних (Data Flow Diagram, DFD), яка відображає логіку взаємодії компонентів у межах розробленої системи. Використання моделі структурного проектування дозволяє візуалізувати процеси обробки та передачі інформаційних потоків у середовищі мобільної операційної системи.

Згідно з методологією структурного аналізу, проектування розпочинається з побудови контекстної діаграми (діаграми нульового рівня), яка визначає межі системи та її інтерфейси із зовнішніми сутностями. У подальшому ця модель деталізується шляхом декомпозиції базових процесів та потоків даних, що забезпечує повне відображення специфіки функціонування комплексу скремблювання та криптографічного захисту цифрових сигналів.



Рисунок 3.3 Діаграма взаємодії процесів

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

Програмно-апаратне скремблювання цифрових сигналів передбачає кероване перетворення послідовності цифрових даних із можливістю її подальшого відновлення на приймальній стороні.

Для систем, що функціонують на базі мобільних операційних систем, необхідно забезпечити узгоджену роботу програмного застосунку, апаратного модуля оброблення сигналів і каналу передавання даних.

Основними завданнями алгоритмічного забезпечення є формування цифрового потоку, його скремблювання, синхронізація передавальної та приймальної сторін, контроль цілісності кадрів і виконання зворотного перетворення. Запропоноване проєктне рішення орієнтоване на створення модульної системи, у якій операції оброблення можуть розподілятися між мобільним пристроєм та зовнішнім апаратним компонентом. Функціонування комплексу доцільно організувати за принципом послідовного проходження цифрових даних через декілька етапів.

На передавальній стороні мобільний застосунок отримує сигнал, перетворює його на послідовність байтів і формує кадри встановленого формату. Після цього виконується скремблювання інформаційного поля та передавання результату через апаратний інтерфейс.

На приймальній стороні здійснюються перевірка кадру, відновлення вихідних даних і передавання їх відповідному програмному компоненту.

Загальну блок-схему функціонування системи передбачено на рисунку 4.1. Вона відображає основні етапи оброблення сигналів та взаємодію програмної й апаратної частин.

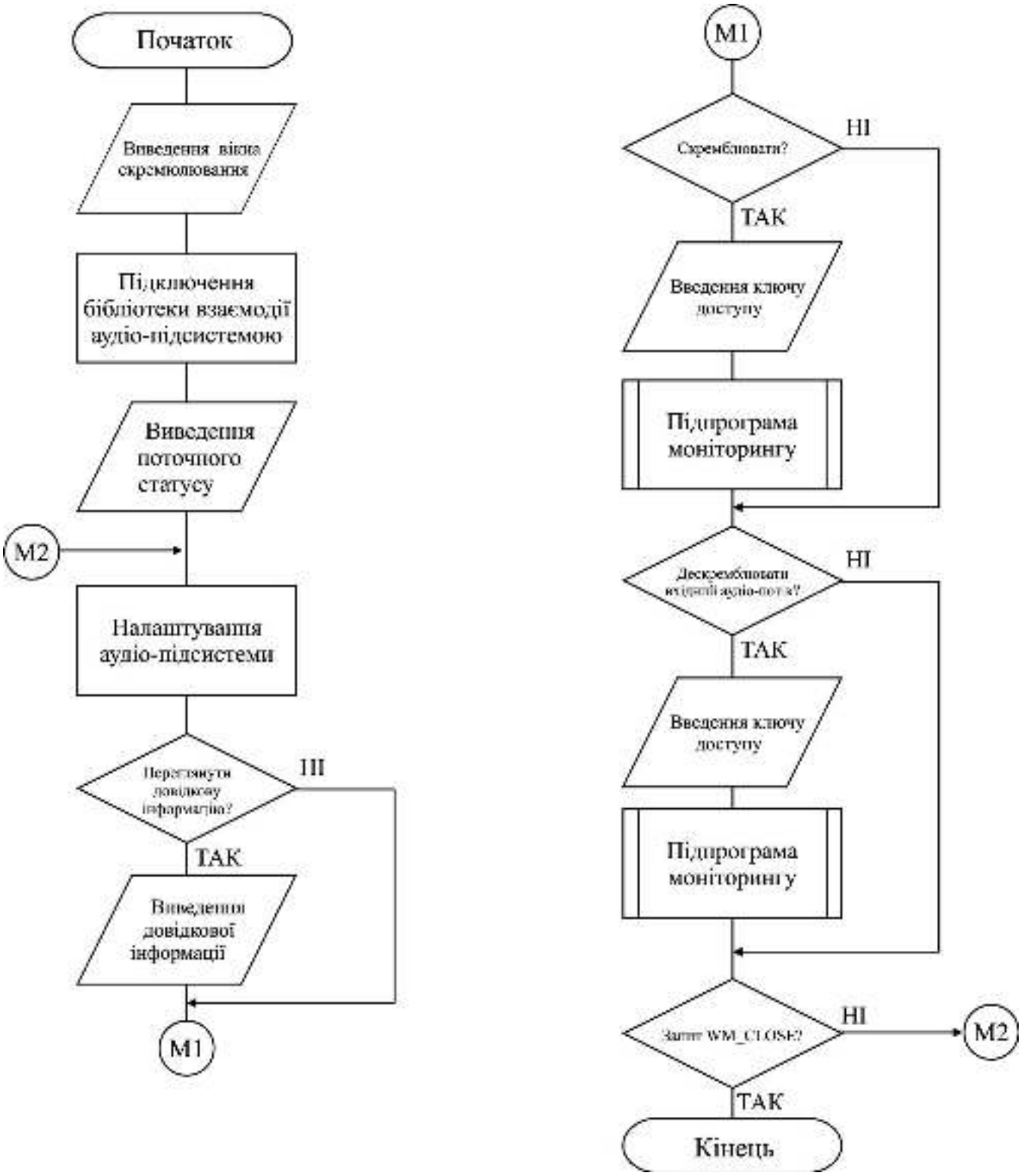

Рисунок 4.1 – Блок-схема основної програми

Вхідними даними системи можуть бути оцифровані аудіосигнали, потоки від датчиків або інша двійкова інформація. Для базової реалізації доцільно використовувати універсальне представлення даних у вигляді масиву байтів.

Це дозволяє відокремити алгоритм скремблювання від фізичної природи сигналу. Додатковими параметрами є ідентифікатор сеансу, номер кадру, довжина інформаційного блока та початковий стан генератора псевдовипадкової послідовності.

Результатом роботи є сформований кадр зі скрембльованими даними та службовими відомостями, необхідними для їх відновлення.

Основною операцією перетворення доцільно обрати синхронне адитивне скремблювання, за якого інформаційна послідовність поєднується з псевдовипадковою бітовою послідовністю за допомогою операції XOR.

Для формування останньої можна застосувати регістр зсуву з лінійним зворотним зв'язком. Такий механізм характеризується відносно простою програмною та апаратною реалізацією.

Окрему блок-схему підпрограми формування скремблювальної послідовності та оброблення інформаційного блока передбачено на рисунку 4.2.

Її призначення полягає у визначенні порядку генерування бітів, виконання зворотного перетворення та контролю умов завершення оброблення.

Обґрунтування побудови програмно-апаратного тракту оброблення сигналів

Архітектуру системи доцільно формувати з урахуванням обмежень мобільних пристроїв щодо доступу до апаратних інтерфейсів, енергоспоживання та тривалості оброблення даних. Мобільна операційна система забезпечує роботу користувацького застосунку, керування джерелами сигналів і взаємодію із зовнішнім обладнанням. Апаратний модуль виконує операції, які потребують стабільної часової організації або безпосереднього підключення до фізичного каналу. Як базовий варіант проєктного рішення можна розглядати мобільний пристрій з Android і зовнішній мікроконтролерний модуль, підключений через USB.

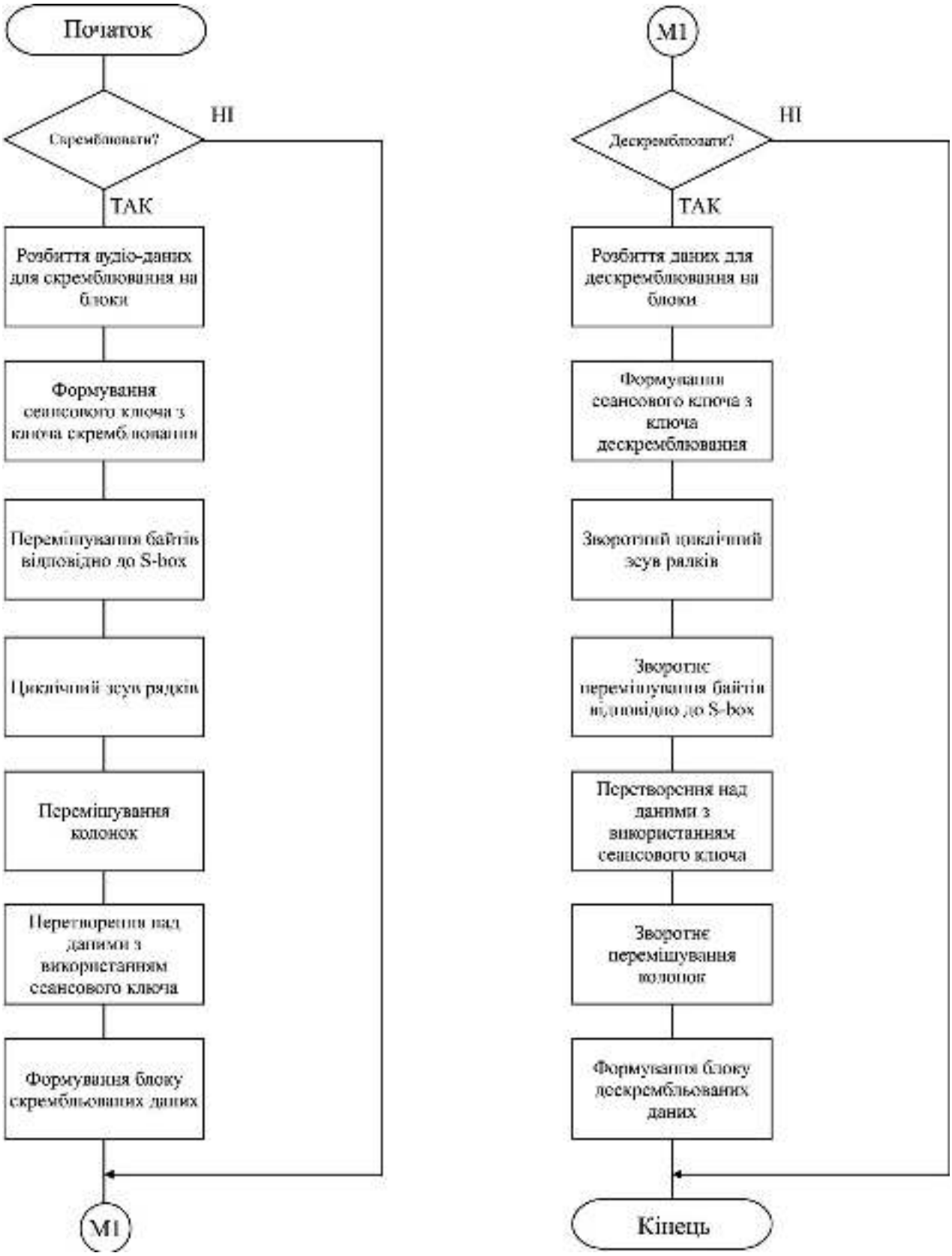

Рисунок 4.2 – Блок-схема роботи підпрограми

Така організація дозволяє використовувати механізми USB Host для обміну даними між застосунком та апаратним пристроєм. Конкретний протокол обміну визначається можливостями мікроконтролера та його USB-реалізації. Підтримка потрібного режиму роботи USB повинна перевірятися для обраної моделі мобільного пристрою.

У запропонованій структурі мобільний застосунок виконує підготовку даних, формування конфігураційних команд, ініціювання сеансу та контроль результатів оброблення. Апаратний модуль може реалізовувати генератор псевдовипадкової послідовності, операції XOR і передавання кадрів до зовнішнього каналу. Використання мікроконтролера не є обов'язковим для самого скремблювання, оскільки аналогічні операції можуть виконуватися програмно. Його застосування обґрунтовується необхідністю безпосередньої роботи з апаратними інтерфейсами та зменшенням залежності часових характеристик оброблення від планувальника мобільної ОС.

Для визначення меж відповідальності компонентів необхідно передбачити декілька режимів функціонування. У програмному режимі скремблювання та дескремблювання виконуються мобільним застосунком. Апаратний модуль використовується лише для передавання або приймання кадрів.

У програмно-апаратному режимі мобільна система формує кадри, а мікроконтролер виконує перетворення інформаційного поля. Обидва режими повинні використовувати однакові параметри генератора та порядок оброблення бітів.

Порівняння зазначених режимів має практичне значення для дослідження продуктивності. Програмна реалізація дозволяє оцінити час перетворення даних без витрат на обмін із зовнішнім обладнанням. Програмно-апаратний режим враховує передавання інформації через USB, виконання операцій мікроконтролером та повернення результату. Вибір оптимального режиму повинен ґрунтуватися на експериментально визначених часових характеристиках.

Для забезпечення незалежності від джерела сигналу доцільно використовувати окремий модуль нормалізації вхідних даних.

У випадку аудіосигналу він приймає відліки визначеної розрядності та формує послідовність байтів. Для сигналів від цифрових датчиків модуль може виконувати перевірку формату повідомлення без додаткового перетворення. Скремблер не повинен змінювати кількість інформаційних бітів або інтерпретувати зміст переданих значень.

Важливо врахувати, що скремблювання та криптографічне шифрування мають різне призначення. Лінійний скремблер може використовуватися для зміни статистичних характеристик цифрової послідовності або узгодження з вимогами каналу передавання. Однак він не забезпечує криптографічної конфіденційності. Якщо система повинна захищати інформацію від навмисного перехоплення, необхідно передбачити додатковий криптографічний механізм. У такому випадку скремблювання виконується як окремий етап оброблення сигналу, а не як заміна шифрування.

Математична модель скремблювання та віднолення цифрової послідовності

Для опису перетворення введемо вихідну двійкову послідовність

$$B = \{b_0, b_1, \dots, b_{(N-1)}\},$$

де N визначає кількість інформаційних бітів, а $b_i \in \{0,1\}$.

Генератор формує псевдовипадкову послідовність

$$G = \{g_0, g_1, \dots, g_{(N-1)}\},$$

яка має відповідати довжині інформаційного блока. Кожен біт скрембльованої послідовності визначається співвідношенням

$$s_i = b_i \oplus g_i,$$

де $\oplus$ позначає додавання за модулем два.

Відновлення даних здійснюється повторним застосуванням тієї самої операції

$$\tilde{b}_i = s_i \oplus g_i.$$

Правильність зворотного перетворення випливає з властивості XOR

$$(b_i \oplus g_i) \oplus g_i = b_i.$$

Отже, за умови використання однакових псевдовипадкових послідовностей передавальна та приймальна сторони отримують тотожні значення інформаційних бітів. Якщо їхні генератори перебувають у різних станах, правильність дескремблювання порушується.

Для формування послідовності пропонується використовувати лінійний регістр зсуву зі зворотним зв'язком LFSR. Його стан представлено двійковим вектором

$$Q_t = (q_{(t,0)}, q_{(t,1)}, \cdots, q_{(t,m-1)}),$$

де m є розрядністю регістра, а t позначає номер кроку.

На кожному кроці обчислюється біт зворотного зв'язку. Він визначається операцією XOR над вибраними розрядами поточного стану. Після обчислення виконується зсув регістра та внесення нового біта. Вихідний біт може зчитуватися з визначеного розряду до оновлення стану.

Для демонстраційної реалізації можна використати 15-розрядний регістр. За умови нумерації розрядів від нуля з молодшого боку його оновлення визначається формулою

$$f_t = q_{(t,14)} \oplus q_{(t,13)},$$

$$Q_{(t+1)} = \left((Q_t \ll 1) \vee f_t\right) \& \left(2^{(15)} - 1\right).$$

Тут $\ll$ позначає побітовий зсув ліворуч, $\vee$ є операцією побітового АБО, а $\&$ визначає побітове І. Вихідний біт зчитується зі старшого розряду перед оновленням регістра.

Для такого варіанта зворотного зв'язку ненульовий початковий стан дозволяє сформувати послідовність із періодом $2^{(15)} - 1 = 32767$ бітів. Нульовий стан використовувати не можна, оскільки він не забезпечує формування змінної псевдовипадкової послідовності. Значення періоду визначає обмеження на довжину фрагмента, який доцільно обробляти без повторного ініціалізування генератора.

У програмно-апаратній системі необхідно забезпечити однакову інтерпретацію стану регістра.

Зокрема, програмна та мікроконтролерна реалізації мають використовувати той самий порядок зсуву, ті самі розряди зворотного зв'язку та однаковий порядок формування байтів. Розбіжність хоча б одного параметра призведе до неправильного відновлення сигналу.

Приклад програмної реалізації скремблера мовою Python наведено нижче. Функція приймає масив байтів і початковий стан генератора. Результатом є послідовність байтів тієї самої довжини.

```
def lfsr_stream(seed: int, bit_count: int):
    # Перевірка допустимого початкового стану
    if not 1 <= seed < (1 << 15):
       raise ValueError("Некоректний стан LFSR")
    state = seed
    for _ in range(bit_count):
        # Формуємо вихідний біт із старшого розряду
        output_bit = (state >> 14) & 1
        # Обчислюємо зворотний зв'язок регістра
        feedback = (
            ((state >> 14) ^ (state >> 13)) & 1
        )
        # Зсуваємо регістр із обмеженням 15 біт
        state = ((state << 1) | feedback) & 0x7FFF
        yield output_bit
def scramble_data(data: bytes, seed: int) -> bytes:
    # Генеруємо скремблювальну послідовність
    generator = lfsr_stream(seed, len(data) * 8)
    result = bytearray()
    for value in data:
        mask = 0
        # Формуємо маску від старшого біта до молодшого
        for _ in range(8):
            mask = (mask << 1) | next(generator)

        # Скремблювання виконується операцією XOR
        result.append(value ^ mask)
    return bytes(result)

def descramble_data(data: bytes, seed: int) -> bytes:
    # Зворотне перетворення використовує той самий LFSR
    return scramble_data(data, seed)
```

Наведені функції ілюструють можливий програмний варіант скремблювання. Для їх застосування необхідно узгодити значення початкового стану з приймальною стороною.

Параметр seed використовується для ініціалізації генератора, а не як криптографічний ключ. Повторне застосування функції до результату першого перетворення з тим самим значенням seed відновлює вихідні дані.

Часова складність алгоритму становить $O(N)$, де N є кількістю оброблюваних бітів.

Для програмної реалізації основні витрати пов'язані з побітовими операціями та формуванням вихідного масиву. У мікроконтролерному варіанті обчислення можуть виконуватися за допомогою стандартних операцій зсуву, XOR та побітових масок.

Додаткова оперативна пам'ять для самого генератора визначається розрядністю регістра.

Попри простоту обчислень, послідовна програмна обробка кожного біта може бути менш ефективною порівняно з формуванням багатобайтових масок або застосуванням апаратних засобів.

Тому доцільно окремо дослідити побітову та блочну реалізації. Їхня функціональна еквівалентність перевіряється порівнянням результатів для однакових вхідних послідовностей і початкових станів.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програмне забезпечення системи програмно-апаратного скремблювання цифрових сигналів призначене для керованого перетворення цифрових потоків, передавання оброблених даних через апаратний модуль та відновлення вихідної інформації на приймальній стороні. Програма забезпечує налаштування параметрів скремблювання, ініціалізацію з'єднання з апаратним пристроєм, оброблення інформаційних кадрів і контроль результатів передавання. Основою функціонування є узгоджена взаємодія мобільної операційної системи та зовнішнього мікроконтролерного модуля. Головне вікно програми показано на рисунку 5.1.

Робоче середовище системи передбачає використання мобільного пристрою, який виконує функції керування сеансом оброблення сигналів. Програмна частина забезпечує приймання цифрових даних, формування кадрів і передавання команд апаратному компоненту. Мікроконтролерний модуль призначений для виконання побітових операцій скремблювання та взаємодії із зовнішнім каналом передавання. Такий розподіл функцій дозволяє відокремити керування конфігурацією від безпосереднього виконання операцій над цифровим потоком.

Для реалізації мобільної частини доцільно використовувати операційну систему Android. Вона надає програмні засоби доступу до периферійних пристроїв та керування їх підключенням. Як мову розроблення мобільного застосунку може бути використано Kotlin. Взаємодія з апаратним модулем організовується через Android USB Host API. Цей механізм дозволяє виявляти підключені USB-пристрої, отримувати дозвіл користувача на доступ і виконувати обмін даними. Конкретний спосіб передавання визначається USB-інтерфейсом, реалізованим у мікроконтролері.

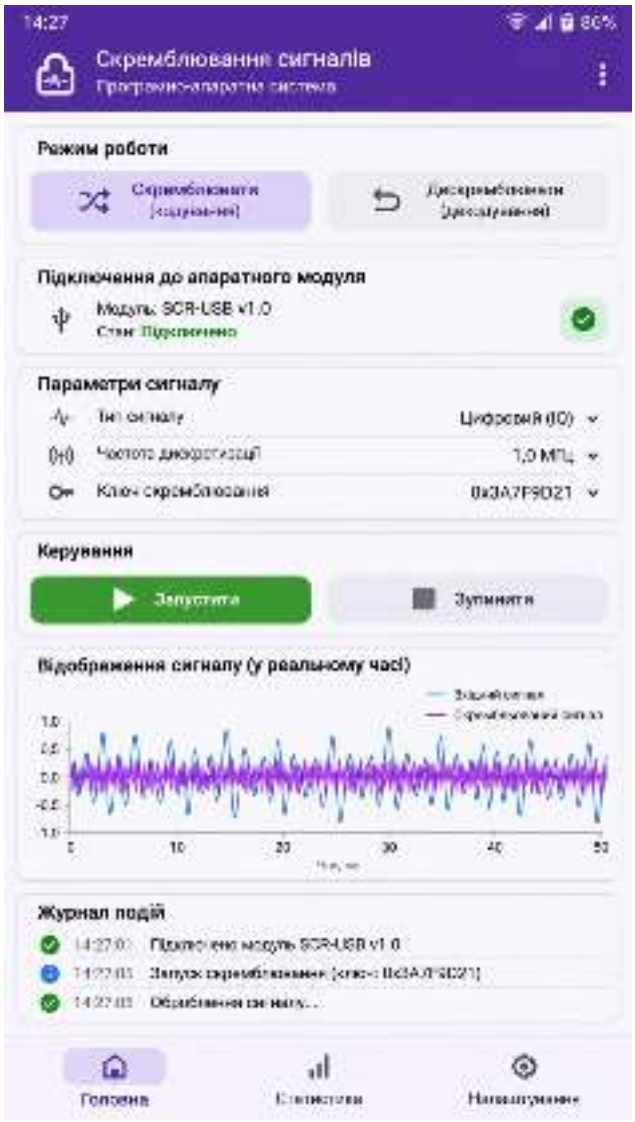

Рисунок 5.1 – Головне вікно розробленого ПЗ

Алгоритмічну частину скремблювання можна реалізувати мовою Python для попереднього моделювання та перевірки правильності перетворення. Відповідні функції генерації псевдовипадкової послідовності та виконання операції XOR розглянуто в попередньому розділі. Під час перенесення алгоритму до мобільного застосунку або мікроконтролера необхідно зберегти однакову логіку оновлення регістра, порядок оброблення бітів та правила формування вихідних байтів. Це забезпечує сумісність програмної й апаратної реалізацій.

Для функціонування комплексу мобільний пристрій повинен підтримувати режим USB Host і забезпечувати можливість підключення зовнішньої периферії.

Апаратний модуль має містити мікроконтролер із достатніми ресурсами для виконання операцій зсуву, XOR та буферизації даних. Його інтерфейс повинен бути сумісним із вибраним способом підключення. Додатково необхідно врахувати вимоги до електроживлення. Якщо мобільний пристрій не забезпечує достатнього струму живлення зовнішнього обладнання, слід використовувати окреме джерело живлення або відповідний USB-концентратор.

Підготовка програмного середовища починається зі встановлення мобільного застосунку та перевірки доступності необхідних системних дозволів. Для розроблення й налагодження Android-застосунку доцільно використовувати Android Studio та відповідні компоненти Android SDK. Перед збиранням програми необхідно перевірити сумісність обраного рівня API з цільовими мобільними пристроями. Функції обміну через USB повинні виконуватися без блокування основного потоку інтерфейсу, оскільки тривале очікування відповіді апаратного модуля може порушувати нормальну роботу застосунку.

Наступним етапом є підготовка мікроконтролерного модуля. Для нього необхідно передбачити програму керування інтерфейсом передавання, приймання конфігураційних команд і виконання скремблювання. Програмування мікроконтролера здійснюється засобами, які відповідають конкретній апаратній платформі. Після завантаження керувальної програми потрібно перевірити ініціалізацію USB-інтерфейсу та можливість обміну службовими повідомленнями. До встановлення зв'язку з мобільним пристроєм апаратний модуль не повинен розпочинати оброблення інформаційного потоку.

Первинне налаштування системи передбачає визначення параметрів скремблювання. Для розглянутої алгоритмічної моделі використовується 15-розрядний регістр зсуву з лінійним зворотним зв'язком LFSR. Перед початком оброблення необхідно встановити допустимий ненульовий початковий стан генератора. Передавальна та приймальна сторони повинні використовувати однакові параметри регістра. Окремо задаються розмір інформаційного блока, порядок формування кадрів і правила синхронізації.

Для передавання цифрових даних доцільно використовувати кадри визначеної структури. Службова частина може містити ідентифікатор сеансу, порядковий номер кадру, довжину інформаційного поля та відомості для синхронізації генератора. Інформаційне поле передається після виконання скремблювання. Для перевірки правильності відновлення можна застосувати контрольну суму CRC, обчислену для вихідних даних перед їх перетворенням. Приймальна сторона перевіряє її після дескремблювання. Конкретний формат кадру має бути однаковим для мобільної та апаратної частин комплексу.

Рекомендована послідовність запуску починається з підключення та ініціалізації апаратного модуля. Після цього запускається мобільний застосунок, який перевіряє наявність пристрою та запитує дозвіл на доступ до нього. Наступним кроком є встановлення каналу обміну й передавання конфігураційних параметрів. Після підтвердження їх прийняття виконується узгодження початкового стану генератора. Лише після успішного завершення зазначених операцій система може переходити до передавання інформаційних кадрів.

Під час роботи програмний модуль формує послідовності байтів із визначеним розміром блока. У програмному режимі перетворення виконується безпосередньо мобільним застосунком. У програмно-апаратному режимі інформаційний блок передається мікроконтролеру для подальшого скремблювання. Результат повертається програмному компоненту або спрямовується до зовнішнього каналу. Для запобігання втратам інформації необхідно передбачити буферизацію та контроль заповнення черги передавання.

Приймальна сторона виконує зворотну послідовність операцій. Спочатку перевіряється коректність структури отриманого кадру та його належність до поточного сеансу. Після відновлення початкового стану генератора виконується дескремблювання інформаційного поля. Отримані дані передаються відповідному програмному компоненту лише після успішної перевірки їхньої цілісності. У разі втрати кадру необхідно забезпечити повторну синхронізацію відповідно до прийнятого формату передавання.

Перевірку працездатності доцільно розпочинати з передавання заздалегідь визначеної тестової послідовності. Спочатку необхідно перевірити правильність програмного скремблювання та подальшого дескремблювання. Отримані дані повинні повністю збігатися з вихідними. Наступним етапом є передавання тієї самої послідовності через апаратний модуль. Порівняння результатів дозволяє встановити відповідність реалізацій алгоритму та правильність обміну через USB.

Додатково необхідно перевірити реакцію системи на відключення апаратного пристрою, порушення синхронізації генераторів і надходження пошкодженого кадру. Для кожної ситуації повинно формуватися відповідне повідомлення про помилку. Після повторного підключення обладнання необхідно відновити сеанс обміну та повторно передати параметри скремблювання. Автоматичне продовження оброблення без перевірки синхронізації не допускається, оскільки це може призвести до неправильного відновлення цифрових даних.

Для оцінювання експлуатаційних характеристик системи рекомендується визначати швидкість оброблення цифрового потоку, затримку передавання кадрів і використання обчислювальних ресурсів мобільного пристрою. Вимірювання необхідно виконувати окремо для програмного та програмно-апаратного режимів. Це дозволяє встановити вплив передавання даних через USB та оброблення мікроконтролером на загальну продуктивність комплексу. Отримані характеристики мають оцінюватися за однакових обсягів інформаційних блоків і параметрів генератора.

Під час подальшої експлуатації необхідно контролювати сумісність версій мобільного застосунку та програми мікроконтролера. Зміни формату кадрів, параметрів генератора або порядку оброблення бітів повинні одночасно враховуватися в обох компонентах. Для діагностики рекомендується вести журнал підключень, помилок синхронізації та результатів оброблення кадрів. При цьому не слід ототожнювати скремблювання з криптографічним захистом, оскільки використаний лінійний генератор не забезпечує криптографічної стійкості.

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

Метою розробки є дослідження та програмна реалізація системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

Об'єктом дослідження є процес програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

Предметом дослідження є методи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

Методи дослідження базуються на методах схемотехніки, методах 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 програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

– Розроблено вітчизняний продукт програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем, який має більш широкі можливості, на відміну від існуючих аналогів.

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

7.1 Вступ

Програмісти у процесі роботи отримують негативний вплив на органи зору, а також мають значну розумову напругу і нервово-емоційне навантаження. Руки (м’язи рук та суглоби пальців) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій спеціалісти відносять високочастотні електромагнітні коливання роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

До недоліків умов праці користувачів комп’ютерної техніки можна віднести:

– недостатню площу і обсяг виробничого приміщення;

– недотримання вимог, мікроклімату на робочих місцях;

– низький рівень освітленості у приміщеннях і на робочих поверхнях апаратури;

– підвищений рівень низькочастотних магнітних полів від моніторів;

– порушення вимог організації робочих місць;

– недотримання вимог до режимам праці та відпочинку;

– надмірне виробничу навантаження працівників;

– відсутність навичок зниження впливу психоемоційного напруги.

Відповідно до ст.14 Закону «Про охорони праці» [3] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування коштів індивідуальної захисту працівників; відповідні вимоги охорони праці, умови праці в кожному робоче місце; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам виконання; інструктаж з охорони праці; організацію

контролю над станом умов праці в робочих місць; проведення атестації робочих місць в умовах праці.

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – ось одна з головних задач охорони праці [5].

7.2 Характеристика умов праці програміста

В приміщенні, в якому проводиться розробка і дослідження програмного продукту, відсутні умови, які можуть створювати підвищену або особливо підвищену небезпеку, тому воно відноситься до класу звичайних приміщень згідно Правил улаштування електроустановок (ПУЕ). Джерелом живлення є трифазна мережа напруги 380/220 В з глухо заземленою нейтралі, з частотою 50 Гц згідно За пожежною та вибухонебезпекою приміщення відноситься до класу В. В таблиці 7.1 наведена загальна характеристика приміщення щодо пожежної та вибухонебезпеки та важкістю робіт.

Таблиця 7.1. – Загальна характеристика приміщення щодо вибухопожежонебезпеки та важкістю робіт

Характеристика приміщень за вибухопожежною категорією та класом зони	Загальна характеристика приміщення	Категорія за важкістю робіт згідно ГН 3.3.5-7.6.6.1 -2002
В – пожежонебезпечне клас П – II	Звичайне без ознак хімічного забруднення та нормальної вологості і за санітарними нормами	1а - до 139 Вт/м^2 1б - до 140-174 Вт/м^2 Клас умов праці – оптимальний

Температура повітря в приміщенні визначається температурою зовнішнього повітря і тепловою енергією, що виділяється всередині

приміщення. Джерелами теплоти в даному приміщенні є люди, електроустаткування, а також освітлювальні прилади в темний час доби. Зовнішнім джерелом надлишкового тепла є сонячна радіація у світлий час доби. Робота, виконувана в даному приміщенні, відноситься до категорії I-а. Людиною в цьому випадку виділяється до 120 ккал теплової енергії в годину. Вологість повітря в приміщенні визначається вологістю атмосферного і видихуваного людьми повітря, а також випарами з поверхні шкіри.

У приміщенні немає виділення шкідливих газів. Тому що в ньому не проводиться монтажних робіт, пайки чи інших робіт, при яких виділяються шкідливі гази.

Для нормалізації параметрів повітряного середовища також періодично здійснюється провітрювання приміщення і вологе прибирання. У всьому будинку діє встановлена загально обмінна витяжна вентиляція.

Раціональне освітлення приміщення сприяє кращому виконанню виробничого завдання і забезпеченню комфорту при роботі. Для забезпечення нормального освітлення застосовуються природне, однобічне, бічне і штучне освітлення, а також сполучене, нормуються згідно ДБН В.2.5-28-2006 Природне і штучне освітлення [6].

За результатами виміру освітленості величина освітленості від системи загального штучного освітлення дорівнює 310 лк, що відповідає вимогам, які пред'являються до даного приміщення.

Основними джерелами шуму на робочих місцях , обладнаних відео дисплейними терміналами, є принтер, сканер факс і обладнання для кондиціювання повітря, в самих відео дисплейних терміналів – вентилятори систем охолодження і трансформатори.

Згідно ДСанПіН 3.3.2.007-98 [6] допустимий еквівалентний рівень шуму для робочого місця програміста складає 50 дБА (акустичних децибела).

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 в нормі. У кімнаті знаходиться ПК з 4-ядерним процесором і 23-дюймовим IPS монітором, а також меблі.

Приміщення має довжину 4м, 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 7.2.

Таблиця 7.2 – 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. 7.2, 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» [7] 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 120ккал теплової енергії за

годину. Оптимальні та фактичні значення параметрів мікроклімату приведені в таблиці 7.3.

По отриманим замірам параметрів мікроклімату можна зробити висновок, що усі 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» [7] ця робота відноситься до V_a розряду зорових робіт. Передбачається використання природного, штучного і змішаного освітлення.

Таблиця 7.3 – 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»

Природне освітлення здійснюється за допомогою вікна, площа якого складає S' = 1,8*1,5 = 2,7 м^2 і є боковим освітленням. У світильниках місцевого і загального освітлення використовуються світлодіодні лампи потужністю 20 Вт із світловим потоком 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, що не має сильного електромагнітного випромінювання і сертифіковані в Україні.

Блок живлення є екранованим і не випускає вищезазначених видів випромінювання.

7.4 Розробка заходів з охорони праці

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці програміста.

Для зменшення шуму в приміщенні пропоную використовувати замість матричного принтера, що створює багато шуму, більш тихий - лазерний принтер.

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 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, проведених відділом охорони праці.

Як міри по зниженню шуму можна запропонувати:

– облицювання стелі і стін звукопоглинаючим матеріалом (знижують шум на 6-8 до);

– екранування робочого місця (постановкою перегородок, діафрагм);

– установка в комп'ютерних приміщеннях устаткування, що робить мінімальний шум;

– раціональне планування приміщення.

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

7.5 Розрахункова частина

При роботі з комп'ютерною та офісною технікою головними причинами ураження електричним струмом є двохфазне дотикання; однофазне дотикання; дотик до корпуса обладнання, яке не проводить струм, але опинилося під напругою; перебування в зоні дії атмосферної або статичної електрики; вхід у зону дії електромагнітного поля.

Правильна організація експлуатації й обслуговування комп'ютерної та офісної техніки, контрольно-вимірювальної апаратури оговорена "Правилами технічної експлуатації електроустановок споживачів і правил техніки безпеки при експлуатації електроустановок споживачів" [8].

Розрахуємо конструкцію захисного заземлення (рис. 7.1) за методикою [9, 10].

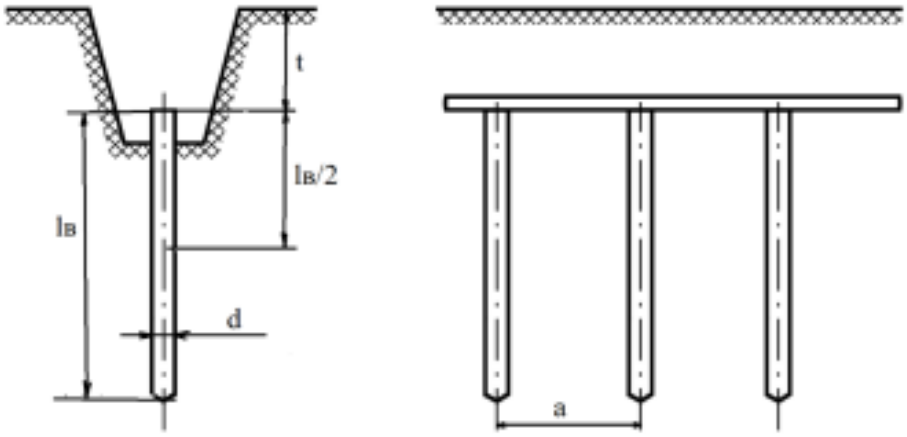

Рисунок 7.1 - Схема електричного захисного заземлення

Початкові дані для розрахунку:

– напруга живлення 380 В;

– режим нейтралі трансформатора – ізольована; – природне заземлення присутнє, R_e = 20 Ом;

– ґрунт – глина; питомий опір – ρ_r = 60 Ом·м;

– вологість глини під час вимірювання її опору – нормальна; – кліматична зона – третя.

Відповідно рекомендацій ПУЕ умови виконання завдання визначають значення

$$R_{nye} = 4 \text{ Ом}.$$

Природні заземлювачі наявні, тому допустимий опір R_d штучних заземлювачів

$$R_d \leq \frac{20 \cdot 4}{20-4} \leq 5 \text{ Ом},$$

Для виконання заземлення обираємо вертикальні електроди із кутової сталі шириною профілю b_k = 50 мм, довжиною l_v = 3 м; для таких вертикальних електродів коефіцієнт сезонності ψ = 1,3;

– відстань між електродами a = 3 м, тобто a / l_v = 1;

– горизонтальний з'єднувальний електрод – стальна смуга шириною b_g = 45 мм;

– глибина закладання смуги t =0.8 м.

Визначаємо розрахунковий питомий електричний опір глини ρ_1, де буде розміщуватися заземлювальний пристрій, для третьої кліматичної зони місцевості $\boldsymbol{\psi}$ = 1.3:

$$\rho_1 = 60 \cdot 1{,}3 = 78 \text{ Ом м.}$$

Визначаємо опір розтіканню струму з одного вертикального заземлювача, попередньо визначивши відстань від поверхні землі до середини вертикального заземлювача $t_{\boldsymbol{v}}$:

$$t_v = 0{,}8 + 0{,}5 \cdot 3 = 2{,}3 \text{ м}$$

$$R_v = \frac{78}{2 \cdot 3.13 \cdot 3} ln \frac{2 \cdot 3}{0.95 \cdot 0.05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2.3 + 3}{4 \cdot 2.3 - 3} = 21.424 \text{ Ом}$$

Розраховуємо кількість вертикальних електродів n методом послідовних наближень. У першому наближенні приймаємо $\eta_v = 1$. Виконуючи округлення результату розрахунку у більший бік отримуємо:

$$n_1 = \frac{21.424}{5} \approx 5.4$$

Для n_1 за таблицею визначаємо η_{v1}. Оскільки значення η_{v1} для n_1 у таблиці відсутнє, знаходимо середнє арифметичне:

$$\eta_{v1} = \frac{0.73 + 0.65}{2} = 0{,}69.$$

Для η_{v1} визначаємо нове число n2, округляємо його до більшого цілого:

$$n_2 = \frac{21.424}{5 \cdot 0{,}69} \approx 7.$$

Для n_2 = 7 визначимо η_{v2}:

$$\frac{n_2 - 6}{10 - 6} = \frac{\eta_{v2} - 0{,}65}{0.59 - 0.65}$$

Для η_{v3} визначаємо нове число $n3$ та округляємо результат до більшого цілого:

$$n_3 = \frac{21{,}424}{5 \cdot 0{,}635} \approx 7.$$

Таким чином отримали, що кількість електродів не змінюється (різниця між останніми числами заземлювачів менше одиниці). Тобто число вертикальних заземлювачів складає 7.

Визначаємо довжину горизонтального електроду. Для заземлювачів, розташованих у ряд

$$L_g = 1{,}05 \cdot 1 \cdot (7 - 1) = 17.9 \text{ м.}$$

Знаходимо опір розтікання струму горизонтального електроду, визначивши спочатку величину ρ_2 , $\boldsymbol{\psi} = 2{,}5$:

$$\rho_2 = 60 \cdot 2{,}5 = 150 \text{ Ом·м,}$$

$$R_g = \frac{150}{2\pi \cdot 18.9} \ln \frac{2 \cdot 18{,}9 \cdot 18.9}{0.045 \cdot 0.8} = 12.5 \text{ Ом}$$

Визначаємо загальний результуючий опір системи заземлення, спочатку визначивши коефіцієнт використання горизонтального електроду за допомогою табличними даними. Для цього, виконуючи лінійну інтерполяцію, напишемо рівняння:

$$\eta_g = - 0{,}00667 \cdot n_3 + 0{,}7933 = 0{,}747;$$

$$R_{gr} = \frac{21{,}424 \cdot 12{,}5}{21.424 \cdot 0.747 + 7 \cdot 12.5 \cdot 0.636} = 3{,}742 \text{ Ом.}$$

Порівнюємо значення R_{gr} з R_d. Співвідношення $R_{gr} < R_d$ виконується, тобто вибрана конструкція заземлення задовольняє вимогам ПУЕ.

Таким чином, приймаємо наступну конструкцію заземлення: вертикальні електроди розміщені в ряд в кількості 7 штук, довжина з’єднувального горизонтального електрода складає 18,9 м.

8 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

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 програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

– Досліджена система програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

– На основі отриманих результатів досліджень створена програмна реалізація системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем.

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. Волошин Є.О. Дослідження та програмна реалізація системи програмно-апаратного скремблювання цифрових сигналів на базі мобільних операційних систем // Збірник праць молодих науковців ЦНТУ. – Вип. 17. – Кропивницький: ЦНТУ, 2027.

2. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.

3. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.

4. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p

5. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.

6. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.

7. 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.

8. 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.

9. 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.

10. 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.

11. 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.

12. 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/

13. 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.

14. 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.

15. 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.

16. 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.

17. 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

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

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

31. 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

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

40. 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,

41. 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.

42. 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.

43. 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.

44. 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.

45. 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.

46. 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

47. 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.

48. 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.

49. 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.*

50. 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.

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

52. 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.

53. 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.

54. 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.

55. 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.

56. 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.

57. 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.

58. 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.