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
“____” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android”

Виконав здобувач вищої освіти
ІІ курсу, групи КІ-24М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Бобровський Д.О.
«____» _____ 2025 р.

Керівник проекту
доктор технічних наук, професор
_____ Коваленко О.В.
«____» _____ 2025 р.
Рецензент _____
_____

м. Кропивницький

АНОТАЦІЯ

Бобровський Д.О. Дослідження та програмна реалізація системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

Метою розробки є дослідження та програмна реалізація системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

Об'єктом дослідження є процес захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

Предметом дослідження є методи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

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

Результат роботи – програмна реалізація системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

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

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

Програма може використовуватися на мобільному пристрої з ОС Android.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, захист даних, GPRS-мережа

ABSTRACT

Bobrovskyi D.O. Research and software implementation of the GPRS network data protection system deployed on devices running OS Android. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the second (master's) level of higher education, software has been developed that is intended for the GPRS network data protection system deployed on devices running OS Android.

The purpose of the development is the research and software implementation of the GPRS network data protection system deployed on devices running OS Android.

The object of the research is the process of protecting GPRS network data deployed on devices running OS Android.

The subject of the research is methods of protecting GPRS network data deployed on devices running OS Android.

The research methods are based on methods of information protection in the network, methods of mathematical statistics, methods of software development.

The result of the work is a software implementation of the GPRS network data protection system deployed on devices running OS Android.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

The program can be used on a mobile device with OS Android.

The program was developed in the Python environment.

Keywords: computer engineering, data protection, GPRS network

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

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

ВСТУП

Актуальність теми. Щоб можна було скористатися перевагами GSM мереж, їх необхідно захистити. Незахищені бездротові мережі відкривають практично необмежений доступ до корпоративної мережі для хакерів і інших зловмисників, які нерідко прагнуть лише одержати безкоштовний доступ в Internet. Відповідно, дані, які передаються по GPRS, потребують захисту.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

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

– Огляд існуючих систем захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

– Дослідження системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

– Програмна реалізація системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

Об'єктом дослідження є процес захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

Предметом дослідження є методи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

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

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

– Удосконалено метод захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

– Розроблено вітчизняний продукт захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

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

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2025 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

GPRS-модем (блок передачі даних) призначений для перетворення послідовних даних в IP-дані або IP-даних на послідовні дані, а також для передачі даних з бездротового термінального пристрою через мережу бездротового зв'язку. У різних застосуваннях IP-адреса центру обробки даних, номер порту та швидкість передачі даних послідовного порту різняться.

GPRS-модеми повинен підтримувати конфігурацію параметрів, а налаштовані параметри зберігаються у внутрішньому пристрої постійної пам'яті (зазвичай FLASH або EEPROM тощо). Після ввімкнення він автоматично встановлює правильний параметр для роботи. Його зручні та гнучкі характеристики широко використовуються в метеорології, гідрології та водних ресурсах, геології та інших галузях промисловості. Зі швидким розвитком речей, використання бездротових мобільних мереж (GPRS, CDMA,3G,4G) проектів буде дедалі більше, як забезпечити безпеку даних, що передаються через GPRS-модеми загального користування, після цього? Особливо в деяких важливих сферах, таких як фінанси, енергетика та інші галузі промисловості.

1.2 Область застосування

Областю застосування є GPRS-мережі. GPRS – це архітектура мережі передачі даних, спеціально розроблена для інтеграції з GSM-мережами, що надає мобільним користувачам доступ до сервісів пакетної передачі даних. У свою чергу, для передачі пакетних даних по радіоканалах GSM/UMTS оператори використовують протокол GTP (GPRS Tunneling Protocol). GTP дозволяє

перетворити радіосигнали від мобільних станцій у пакети даних і потім передати їх по незашифрованих тунелях через магістраль оператора.

Використовуючи технологію GPRS, мобільні оператори одержали можливість надавати сервіси доступу до інтернету й побудови корпоративних VPN, на додаток до існуючого сервісу передачі голосу.

У загальному випадку основними елементами GPRS-мережі є:

1. BSS (Base Station) – приймає й розпізнає радіосигнал (голос або дані) і транслює GPRS-дані на SGSN (у випадку передачі даних).

2. SGSN (Serving GPRS Support Node) – вузол обслуговування абонентів; забезпечує підключення нового абонента в мережі й передачу даних до GGSN.

3. GGSN (Gateway GPRS Support Node) – вузол маршрутизації GPRS, приймає й передає дані із зовнішніх мереж (Інтернет, мережі абонентів, партнерів), а також, видає IP-адреси абонентам і тарифікує їхньої послуги (у взаємодії із системою біллінгу).

4. BG (Border Gateway) – прикордонний шлюз, що використовується для зв'язку з PLMN інших операторів.

GGSN і SGSN усередині мережі одного оператора (local PLMN) взаємодіють через протокол GTP. Для спрощення цей зв'язок називають інтерфейс Gn. Підключення між мережами (PLMN) різних операторів формують інтерфейс Gp, а вихід у зовнішні мережі (клієнтів, Інтернет) – інтерфейс Gi. І нарешті, інтерфейс Ga – служить для підключення GPRS-мережі до внутрішньої біллінгової системи.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

1. Мобільна безпека Lookout

Огляд: Lookout Mobile Security пропонує захист від мобільних загроз на базі штучного інтелекту, виявлення фішингу та безпеку даних для підприємств.

Переваги:

- Виявлення загроз за допомогою штучного інтелекту.
- Надійний захист від фішингу.

Недоліки:

- Це може бути дорого для малого бізнесу.
- Обмежена функціональність офлайн.

Оцінки користувачів:

- Рейтинг G2: 4.3/5 (69 відгуків).
- Рейтинг Gartner: 4.6/5 (92 відгуки).

2. Zimperium MTD

Огляд: Zimperium MTD забезпечує виявлення шкідливих програм на основі машинного навчання та загроз нульового дня, забезпечуючи мобільну безпеку в режимі реального часу.

Переваги:

- Надійне виявлення загроз нульового дня.
- Аналітика безпеки на основі штучного інтелекту.

Недоліки:

- Потрібне навчання для оптимального налаштування.
- Вищі ціни для малого та середнього бізнесу.

Оцінки користувачів:

- Рейтинг G2: 4/5 (1 відгук).
- Рейтинг Gartner: 4.4/5 (53 відгуки).

3. IBM MaaS360

Огляд: IBM MaaS360 пропонує мобільну безпеку та єдине керування кінцевими точками (UEM) для підприємств, допомагаючи ефективно захищати мобільні пристрої.

Переваги:

- Комплексні можливості UEM.
- Інтегрований захист від загроз.

Недоліки:

- Складний процес розгортання.
- Преміум-функції вимагають планів вищого рівня.

Оцінки користувачів:

- Рейтинг G2: 4.2/5 (173 відгуки).
- Рейтинг Gartner: 4.4/5 (297 відгуків).

4. Microsoft Defender для кінцевих точок (мобільний)

Огляд: Microsoft Defender для кінцевих точок забезпечує захист мобільних пристроїв підприємства за допомогою вбудованої аналітики загроз.

Переваги:

- Безшовна інтеграція з екосистемою Microsoft.
- Надійний захист від фішингу та шкідливих програм.

Недоліки:

- Потрібна ліцензія Microsoft.
- Обмежена підтримка для середовищ, відмінних від Microsoft.

Оцінки користувачів:

- Рейтинг G2: 4.4/5 (306 відгуків).
- Рейтинг Gartner: 4.0/5 (9 відгуків).

5. Мобільна безпека McAfee

Огляд: McAfee Mobile Security пропонує захист від крадіжки, шифрування та безпечний перегляд веб-сторінок для захисту мобільних пристроїв підприємства.

Переваги:

- Потужні можливості шифрування.
- Функції безпечного перегляду веб-сторінок.

Недоліки:

- Для деяких функцій потрібна преміум-підписка.
- Вплив на продуктивність на старіших пристроях.

Оцінки користувачів:

- Рейтинг G2: 4.0/5 (45 відгуків).
- Рейтинг Gartner: 4.6/5 (40 відгуків).

6. Мобільний захист кінцевих точок Symantec

Огляд: Symantec Endpoint Protection Mobile надає підприємствам аналітику безпеки на основі штучного інтелекту та захист від мобільних загроз.

Переваги:

- Виявлення ризиків за допомогою штучного інтелекту.
- Аналітика безпеки корпоративного рівня.

Недоліки:

- Потребує цілеспрямованого управління.
- Вищі витрати для малих організацій.

Оцінки користувачів:

- Рейтинг G2: 4.1/5 (45 відгуків).
- Рейтинг Gartner: 4.2/5 (73 відгуки).

7. Sophos Intercept X для мобільних пристроїв

Огляд: Sophos Intercept X пропонує безпечний мобільний доступ і захист кінцевих точок завдяки виявленню загроз на основі штучного інтелекту.

Переваги:

- Надійний захист кінцевих точок.
- Доступні ціни.

Недоліки:

- Потрібні знання адміністратора.
- Інтерфейс користувача може бути складним.

Оцінки користувачів:

- Рейтинг G2: 4.6/5 (447 відгуків).
- Рейтинг Gartner: 4.5/5 (35 відгуків).

8. Check Point Harmony Mobile

Огляд: Check Point Harmony Mobile забезпечує мобільну безпеку на основі штучного інтелекту та захист за принципом нульової довіри для бізнесу.

Переваги:

- Розширена безпека на основі штучного інтелекту.
- Захист мобільних пристроїв за принципом нульової довіри.

Недоліки:

- Дорого для невеликих команд.
- Управління складною політикою.

Оцінки користувачів:

- Рейтинг G2: 4.5/5 (44 відгуки).
- Рейтинг Gartner: 4.6/5 (84 відгуки).

9. Мобільна безпека Trend Micro

Огляд: Trend Micro Mobile Security надає функції захисту від шкідливих програм, віддаленого стирання та зашифрованого сховища для захисту підприємств.

Переваги:

- Надійний захист від шкідливих програм.
- Зашифроване сховище.

Недоліки:

- Може уповільнювати роботу пристроїв.
- Вимагає частих оновлень.

Оцінки користувачів:

- Рейтинг G2: 4.0/5 (28 відгуків).
- Рейтинг Gartner: 4.5/5 (22 відгуки).

10. BlackBerry Protect (Cylance Mobile)

Огляд: BlackBerry Protect пропонує мобільну безпеку на базі штучного інтелекту для підприємств, зосереджуючись на проактивному запобіганні загрозам.

Переваги:

- Надійна безпека на основі штучного інтелекту.
- Захист, орієнтований на підприємства.

Недоліки:

- Обмежена інтеграція зі сторонніми розробниками.
- Вища крива навчання.

Оцінки користувачів:

- Рейтинг G2: 4.3/5 (57 відгуків).
- Рейтинг Gartner: 4.7/5 (77 відгуків).

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це об’єктно-орієнтована мова програмування високого рівня загального призначення з відкритим кодом. Це визначення може бути важким для новачків, тому розглянемо кожну характеристику окремо, щоб зрозуміти, що вона означає:

– Відкритий вихідний код: це безкоштовно та доступно для подальших покращень, таких як додавання корисних функцій або виправлення помилок.

– Об’єктно-орієнтована: заснована не на функціях, але в об’єктах з певними атрибутами й методами.

– Високий рівень: зручний для людини, а не для комп’ютера.

– Загальне призначення: можна використовувати для створення будь-яких програм.

Ця мова використовується в будь-якому програмному забезпеченні, про яке ви тільки можете подумати. Ви можете використовувати його для створення веб-сайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу та багато іншого. Також застосовується в науці про дані, аналізі даних, машинному навчанні, інженерії даних, веб-розробці, розробці програмного забезпечення та інших галузях.

Переваги та недоліки Python

Переваги:

– Її легко читати, вчити та писати. Це мова програмування високого рівня з англійським синтаксисом. Це полегшує читання та розуміння коду. Її дійсно легко зрозуміти і вивчити, тому багато людей рекомендують Python новачкам. Вам потрібно менше рядків коду для виконання того ж завдання в порівнянні з іншими основними мовами, такими як C/C++ та Java.

– Підвищує продуктивність. Це дуже продуктивна мова. Завдяки її простоті розробники можуть зосередитися на розв’язанні проблеми. Їм не потрібно витрачати багато часу на розуміння синтаксису або поведінку мови програмування. Ви пишете менше коду та виконуєте більше завдань.

– Інтерпретована мова. Python мова, що інтерпретується, а це означає, що вона безпосередньо виконує код по рядку. Якщо сталася помилка, вона зупиняє подальше виконання та повідомляє про її виникнення. Вона показує лише одну помилку, навіть якщо у програмі їх кілька. Це спрощує налагодження.

– Динамічно типізована. Python не визначає тип змінної, доки ми не запустимо код. Вона автоматично надає тип даних, коли відбувається процес

виконання. Фахівець може не турбуватися про оголошення змінних та типи даних.

– Безкоштовна та з відкритим вихідним кодом. Ця мова постачається під схваленою OSI ліцензією з відкритим вихідним кодом. Це робить його безкоштовним для використання та розповсюдження. Ви можете завантажити вихідний код, змінити його та навіть розповсюджувати свою версію. Це корисно для організацій, які хочуть використати свою версію для розробки.

– Підтримка великих бібліотек. Стандартна бібліотека Python є величезною, ви можете знайти майже всі функції, необхідні для вашого завдання. Таким чином ви не залежите від зовнішніх бібліотек.

– Портативність .У багатьох мовах, таких як C/C++, потрібно змінити свій код, щоб запустити програму на різних платформах. З Python все інакше. Ви тільки пишете один раз і запускаєте її будь-де.

Недоліки:

– Низька швидкість. Вище ми обговорювали, що це інтерпретована мова з динамічною типізацією. Порядкове виконання коду часто призводить до повільного виконання. Динамічна природа Python також є причиною її низької швидкості, оскільки їй доводиться виконувати додаткову роботу при виконанні коду. Тому вона не підходить для цілей, де швидкість важливий аспект проєкту.

– Неефективна для пам’яті. Ця мова програмування використовує великий обсяг пам’яті, це може бути недоліком при створенні програм, коли віддають перевагу оптимізації пам’яті.

– Слабка у мобільних обчисленнях. Python зазвичай використовується у серверному програмуванні. Ми не бачимо – її на стороні клієнта або в мобільних програмах з таких причин: вона не заощаджує пам’ять і має повільну обчислювальну потужність у порівнянні з іншими мовами.

– Доступ до бази даних. Програмувати на цій мові легко, але коли ми взаємодіємо з базою даних, її не вистачає. Рівень доступу до бази даних у Python

примітивний та недостатньо розвинений у порівнянні з іншими популярними технологіями.

– Помилки виконання. Це мова з динамічною типізацією, тому тип даних змінної може змінюватись у будь-який час. Змінна, що містить ціле число, у майбутньому може містити рядок, що може призвести до помилок виконання.

Застосування Python:

– Для аналізу даних. Дані стали цінним активом у будь-якій сучасній галузі, і більшість компаній зацікавлені у збиранні, обробці та аналізі релевантних даних, щоб витягти з них цінну інформацію для бізнесу. І тут Python виходить за межі будь-якої конкуренції. Python особливо цінна тим, що крім великої стандартної бібліотеки надає величезний набір додаткових модулів, розроблених спеціально для аналітичних цілей. Найвідоміші бібліотеки Python для аналізу даних – це pandas і NumPy . Ці інструменти дозволяють робити з вашими даними майже все, наприклад, очищати і аналізувати їх, вивчати статистику або візуалізувати приховані тенденції у ваших даних.

– Для візуалізації даних. Візуалізація даних – це окрема частина аналізу даних, яка допомагає нам подавати інформацію, необроблену чи очищену, у більш змістовній формі. Тут Python знову входить у гру, пропонуючи широкий спектр інструментів візуалізації даних. Найпопулярніші з них – matplotlib і заснований на ній seaborn. Використовуючи їх, ми можемо створювати буквально всі види візуалізації: від найпростіших до складніших.

– Для машинного навчання. Машинне навчання (ML) є основою більшості завдань науки даних. Він є областю штучного інтелекту, пов’язаною з використанням алгоритмів, що дозволяють машинам вивчати закономірності та тенденції на основі історичних даних, щоб робити прогнози на основі невідомих даних. – Використовуючи методи ML, ми можемо створювати моделі, які можуть точно передбачити швидкість відтоку клієнтів компанії, оцінити ризик виникнення у людини певного захворювання, визначити оптимальне

розташування автомобілів таксі й т.д. За допомогою Python ми можемо побудувати модель ML, використовуючи лише три рядки коду.

– Для розробки програмного забезпечення. Крім свого багатостороннього застосування в галузях науки про дані, Python використовується на кожному етапі розробки програмного забезпечення, включаючи контроль складання, автоматичну безперервну компіляцію, прототипування, відстеження помилок, тестування та обслуговування програмного забезпечення. За допомогою цієї мови можемо створювати аудіо- або відеопрограми на основі методів штучного інтелекту, машинного навчання, API (інтерфейсів прикладного програмування), GUI (графічних інтерфейсів) або будь-якого іншого типу програмного забезпечення.

– Для веброзробки. У той час як для створення візуальної частини вебсайту ми переважно будемо використовувати такі мови, як HTML, CSS та JavaScript, для його невидимої частини ми часто вибираємо Python. Серед масштабних вебсайтів та програм, створених за допомогою цієї мови, варто згадати Google, Facebook, Instagram, YouTube, Dropbox та Reddit.

– Для автоматизації задач/скриптингу. Це відмінний інструмент для написання програм для автоматизації різних завдань, що повторюються. Цей процес називається скриптингом. Зокрема, можна робити скрипти для роботи з файлами та папками. Наприклад, можна створювати, перейменовувати, перетворювати, розділяти, об’єднувати або видаляти файли, перевіряти їх на наявність помилок. Ви також можете використовувати автоматизацію Python для пошуку та завантаження інформації з Інтернету, заповнення та надсилання онлайн-форм та надсилання регулярних повідомлень або електронних листів.

Яким фахівцям потрібно володіти Python:

– Фахівець з даних.

– Аналітик даних.

– Інженер даних.

– Інженер з машинного навчання.

– Журналіст даних.

– Архітектор даних.

– Повний стек веб-розробника.

– Backend-розробник.

– DevOps-інженер.

– Інженер-програміст.

Можемо зробити висновок, що Python ще довго буде популярною мовою, хоч і має низку недоліків. Цю мову використовують для створення вебсайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу, аналізу даних, машинного навчання, інженерії даних та для багатьох інших областей. Це перспективна і затребувана навичка, яка необхідна у всіх галузях.

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 захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

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

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

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

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

г) організувати інтерфейс користувача з метою формування та виводу на

екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Безпека мобільних даних є критично важливим пріоритетом у 2025 році, оскільки компанії стикаються зі зростанням кіберзагроз, спрямованих на смартфони, планшети та інші підключені пристрої. Зростання мобільних кіберзагроз, витоків даних та атак шкідливого програмного забезпечення вимагає надійних заходів безпеки. Такі фактори, як віддалена робота, політика BYOD (Bring Your Own Device) та хмарні мобільні сервіси, ще більше ускладнюють проблеми безпеки. Організації повинні впроваджувати рішення для безпеки мобільних даних, щоб запобігти несанкціонованому доступу, витокам даних та фішинговим атакам.

Розуміння безпеки мобільних даних

Безпека мобільних даних передбачає захист конфіденційних даних на мобільних пристроях за допомогою шифрування, автентифікації та методів зменшення загроз.

Забезпечення безпеки мобільних даних має вирішальне значення для запобігання:

– Несанкціонований доступ та витоки даних.

– Атаки програм-вимагачів та шкідливих програм.

– Внутрішні загрози та спроби фітингу.

Як розвивається ландшафт мобільних загроз

Кібератаки продовжують зростати як за частотою, так і за винахідливістю, і нещодавні звіти лідерів галузі, таких як Verizon та IBM, відзначають стабільне зростання з року в рік. Сьогоднішні зловмисники не просто закидають широкі сітки – вони використовують складні методи, спеціально спрямовані на мобільні пристрої та їхніх користувачів.

Нові та постійні тенденції у сфері мобільних загроз включають:

– Розширене шкідливе програмне забезпечення та фішинг: Шкідливі програми та оманливі посилання можуть проникнути на ваш пристрій, видаючи себе за легітимні завантаження або повідомлення. Зловмисники використовують дедалі переконливіші тактики соціальної інженерії, щоб обманом змусити користувачів відмовитися від облікових даних або встановити шкідливе програмне забезпечення.

– Атаки типу «людина посередині»: пристрої в незахищених мережах вразливі до перехоплення зловмисниками їхніх повідомлень. Цей метод може призвести до витоку конфіденційних даних під час чогось такого простого, як перевірка електронної пошти в громадському Wi-Fi.

– Компрометація пристрою: Зловмисники використовують такі методи, як джейлбрейк та рутінг, для обходу обмежень безпеки, що надає їм глибший доступ до пристрою та його даних. Сучасні загрози здатні маскувати їхню активність, що ще більше ускладнює виявлення.

– Вразливості ОС: Зловмисники швидко використовують нещодавно виявлені вразливості в мобільних операційних системах (відстежуються як CVE), іноді швидше, ніж встигають випустити патчі.

– Контроль доступу до корпоративних ресурсів: Організації спостерігають дедалі більше спроб заражених пристроїв підключитися до внутрішніх мереж, що підвищує ризик витоку даних та несанкціонованого доступу до критично важливих для бізнесу активів.

Щоб випередити ці загрози, потрібні багаторівневі захисні рішення та постійне навчання, що гарантує готовність як пристроїв, так і користувачів до будь-яких загроз, які підкине їм ландшафт мобільних загроз, що розвивається.

Ключові проблеми безпеки в мобільних середовищах:

– Ризики незахищеного Wi-Fi: публічні мережі наражають користувачів мобільних телефонів на кіберзагрози.

– Шкідливі програми та мобільне програмне забезпечення: Зловмисники

атакують пристрої через неперевірені програми.

– Ризики втрати та витоку даних: втрачені/викрадені пристрої можуть поставити під загрозу конфіденційні дані.

– Прогалини в безпеці BYOD та корпоративних пристроїв: балансування безпеки з продуктивністю користувачів.

Виявлення та запобігання розширеному джейлбрейку та рутінгу

Надійні рішення для мобільної безпеки використовують низку стратегій для виявлення та протидії складним спробам джейлбрейка (iOS) та рутування (Android) пристроїв. Ці інструменти постійно контролюють цілісність пристроїв шляхом:

– Сканування на наявність незвичайних модифікацій системи або несанкціонованих змін у файлах ОС.

– Виявлення наявності сторонніх інструментів керування, які часто використовуються для джейлбрейка/рутингу.

– Використання поведінкової аналітики для позначення підозрілої активності та підвищення привілеїв.

– Забезпечення автоматичного виконання дій відповідності, таких як обмеження доступу до пристроїв або запуск віддаленого стирання, у разі виявлення потенційної компрометації.

Провідні платформи, такі як Lookout, Zimperium та Symantec, інтегрують ці заходи виявлення загроз, щоб забезпечити доступ до конфіденційних бізнес-ресурсів лише безпечним пристроям, що відповідають політикам.

Вплив генеративного штучного інтелекту на мобільну безпеку

Генеративний штучний інтелект швидко трансформує ландшафт мобільних загроз, впроваджуючи нові та складні методи атак, які вимагають нашої уваги. Від переконливих фішингових повідомлень, що генеруються на льоту, до автоматизованих атак соціальної інженерії, загрози на базі штучного інтелекту тепер можуть обходити традиційні засоби захисту з безпрецедентною швидкістю та тонкістю.

Щоб випереджати ці ризики, що змінюються, організаціям слід:

– Впроваджуйте виявлення загроз на основі штучного інтелекту: використовуйте інструменти безпеки від таких постачальників, як Palo Alto Networks та Symantec, які використовують машинне навчання для виявлення незвичайної поведінки та нових загроз.

– Навчайте користувачів шахрайству на основі штучного інтелекту: регулярно навчайте співробітників та користувачів мобільних пристроїв розпізнавати дипфейкові дзвінки, голосовий фішинг (вішинг) та ретельно розроблені шкідливі повідомлення.

– Впроваджуйте безперервне сканування програм: використовуйте рішення, які відстежують шкідливі програми та код, створений або розроблений за допомогою методів генеративного штучного інтелекту.

– Посилення контролю автентифікації: поєднання біометричної автентифікації та адаптивної багатофакторної автентифікації (MFA) для мінімізації ризику підробки облікових даних за допомогою штучного інтелекту або видання себе за іншу особу.

Інвестування в рішення, що розвиваються разом із загрозами, спричиненими штучним інтелектом, може бути вирішальним фактором між реагуванням на порушення та його запобіганням. Проактивна адаптація зараз є важливою для сучасної мобільної безпеки.

Як виявляються та управляються вразливості операційної системи

Програмне забезпечення для захисту мобільних даних проактивно сканує пристрої на наявність вразливостей операційної системи, постійно відстежуючи нові виявлені CVE (поширені вразливості та ризики). Провідні рішення використовують оцінку ризиків у режимі реального часу, регулярно перевіряючи налаштування пристроїв, конфігурації системи та дозволи. Розширені інструменти від таких постачальників, як Symantec та Lookout, сповіщають адміністраторів щоразу, коли вони виявляють ознаки компрометації, несанкціоновані зміни конфігурації або спроби рутування чи джейлбрейка

пристрою.

Ці рішення допомагають IT-командам випереджати загрози завдяки:

– Виявлення та позначення невиправлених вразливостей і ризикованих конфігурацій.

– Надання автоматизованих інструкцій або опцій віддаленого виправлення.

– Блокування спроб компрометації до того, як відбудеться експлуатація.

Підтримуючи постійний контроль за станом операційної системи та забезпечуючи дотримання надійних конфігурацій, організації можуть значно знизити ризик успішних мобільних атак або витоків даних.

Ключові характеристики, які слід шукати в програмному забезпеченні для безпеки мобільних даних

– Розширений захист від мобільних загроз (MTD): виявлення фішингу, шкідливого програмного забезпечення та загроз нульового дня на основі штучного інтелекту.

– Наскрізне шифрування: Безпечний мобільний зв'язок, електронна пошта та обмін файлами.

– Безпечна контейнеризація: ізоляція бізнес-даних та особистих даних на одному пристрої.

– Можливості віддаленого стирання та блокування: запобігання втраті даних у разі крадіжки або компрометації пристрою.

– Багатофакторна автентифікація (MFA) та біометричний захист: забезпечення безпечного входу.

– Запобігання втраті даних (DLP): обмеження несанкціонованого доступу до даних та їх передачі.

– Мобільний VPN та мережева безпека: захист даних у публічних та корпоративних мережах.

– Хмарне керування мобільною безпекою: керування політиками безпеки для віддалених користувачів.

Інфраструктура мережевої безпеки: попередження нових загроз

Інфраструктура безпеки мобільних мереж відіграє ключову роль у допомозі підприємствам проактивно захищатися від нових і постійно зростаючих загроз. Використовуючи передові засоби захисту на пристроях, подібні до тих, що містяться в провідних пакетах безпеки для настільних комп'ютерів, організації можуть забезпечити безпеку мобільних кінцевих точок, навіть коли користувачі підключаються через незахищені мережі Wi-Fi або стільникового зв'язку.

Деякі ключові переваги включають:

– Виявлення загроз у режимі реального часу: Розширена система безпеки на рівні мережі постійно відстежує шкідливу активність, включаючи спроби фішингу, підозрілий трафік та атаки типу «людина посередник», зупиняючи загрози до того, як вони вплинуть на пристрої або конфіденційні дані.

– Безшовна інтеграція: Ці рішення працюють разом з існуючими політиками безпеки підприємства, поширюючи на мобільні пристрої той самий надійний захист, який уже діє для традиційних кінцевих точок.

– Захист з будь-якого місця: Співробітники часто підключаються з аеропортів, готелів або кав'ярень. Надійний рівень мережевої безпеки гарантує захист конфіденційних бізнес-даних незалежно від місцезнаходження.

Впроваджуючи комплексні стратегії мережевої безпеки від таких постачальників, як Lookout, Zimperium та Symantec, організації можуть значно зменшити свій вплив як відомих, так і нових кіберзагроз на мобільних платформах.

Блокування заражених пристроїв з корпоративних ресурсів

Провідні платформи безпеки використовують комбінацію моніторингу стану пристроїв у режимі реального часу та динамічного контролю доступу, щоб захистити конфіденційні дані компанії від скомпрометованих пристроїв. Коли виявляються загрози – будь то шкідливе програмне забезпечення, фішингові атаки чи неавторизовані програми – програмне забезпечення автоматично

скасовує або обмежує доступ до корпоративних мереж, хмарних сервісів та критично важливих для бізнесу програм.

Наприклад, такі рішення, як Lookout та Zimperium, використовують оцінки ризиків на основі штучного інтелекту для постійного сканування пристроїв. Якщо виявлено загрозу, платформа може:

– Миттєво ізолюйте пристрій від корпоративних VPN, електронної пошти та інструментів для співпраці.

– Застосування політик умовного доступу через інтеграцію з постачальниками ідентифікаційних даних (таких як Okta або Microsoft Entra).

– За потреби активувати протоколи віддаленого стирання.

Поєднуючи автоматичне виявлення, забезпечення дотримання політик та сегментацію на рівні мережі, ці платформи допомагають забезпечити доступ до ресурсів вашої організації лише справних пристроїв, що відповідають вимогам.

Переваги єдиного захисту кінцевих точок в одному клієнті

Керування безпекою мобільних даних для сучасних віддалених працівників може швидко перетворитися на жонглювання десятками інструментів, кожен з яких має свої особливості та портали. Саме тут на допомогу приходить універсальний клієнт захисту кінцевих точок, який поєднує платформи захисту кінцевих точок (EPP), засоби виявлення та реагування на кінцеві точки (EDR) та розширене виявлення та реагування (XDR) в одному оптимізованому пакеті.

Чому варто об'єднати кошти до одного клієнта:

– Спрощене керування безпекою: маючи все під одним дахом, IT-команди можуть контролювати, налаштовувати та застосовувати політики з централізованої панелі інструментів, що зменшує адміністративні труднощі.

– Зменшення складності та покращення продуктивності: Запуск кількох агентів безпеки часто призводить до уповільнення роботи системи та проблем сумісності. Уніфікований клієнт потребує менше ресурсів та зменшує ризик конфліктів.

– Комплексна видимість загроз: інтеграція EPP, EDR та XDR забезпечує швидше виявлення загроз та реагування на них на кінцевих точках, у мережах та хмарних середовищах, чого важко досягти за допомогою об'єднаних інструментів.

– Стабільне покриття: Від програм-вимагачів до шкідливих програм нульового дня, єдине консолідоване рішення гарантує відсутність прогалин у безпеці між непов'язаними інструментами.

– Спрощені оновлення та виправлення: Замість того, щоб витрачати час на оновлення для кількох окремих програм, єдиний клієнт дозволяє розгортати найновіші засоби захисту одночасно, забезпечуючи стійкість усіх кінцевих точок до нових загроз.

Примітно, що провідні постачальники рішень безпеки, такі як CrowdStrike, SentinelOne та Sophos, застосували цей підхід, зробивши надійну комплексну безпеку кінцевих точок новим стандартом для гнучких та безпечних мобільних працівників.

Найкращі практики для посилення мобільної безпеки:

– Впроваджуйте стратегію захисту від мобільних загроз (MTD) – захищайтеся від шкідливого програмного забезпечення, фішингу та мережевих загроз.

– Забезпечте надійне мобільне шифрування та автентифікацію – використовуйте біометричні дані для входу та багатофакторну автентифікацію.

– Розгортання політик безпеки мобільних пристроїв підприємства – визначення правил безпечного використання програм і доступу до даних.

– Моніторинг та усунення ризиків мобільної безпеки – постійне відстеження вразливостей та неправильних конфігурацій.

– Увімкніть безпечний перегляд мобільних веб-сторінок та захист електронної пошти – запобігайте фішинговим атакам та атакам типу «людина посередині» (MITM).

– Забезпечте запобігання втраті даних (DLP) на мобільних пристроях –

контролюйте обмін файлами та обмеження доступу.

Як забезпечити душевний спокій під час вибору програмного забезпечення для мобільної безпеки

Вибір правильного програмного забезпечення для мобільної безпеки має вирішальне значення для захисту ваших особистих даних та забезпечення душевного спокою. Ось як споживачі можуть прийняти обґрунтоване рішення:

1. Широко дослідіть: почніть з вивчення різних варіантів на ринку. Шукайте відгуки та експертні оцінки на надійних технологічних веб-сайтах. Ці дані можуть виявити сильні та слабкі сторони програмного забезпечення.

2. Порівняння функцій: Визначте основні функції, такі як захист від шкідливого програмного забезпечення, можливості захисту від фішингу та запобігання крадіжкам. Порівняйте продукти, щоб побачити, який з них пропонує повне покриття.

3. Перевірте незалежні тести: Шукайте програмне забезпечення, яке пройшло незалежне тестування у відомих лабораторіях кібербезпеки. Сертифікати від сторонніх організацій можуть гарантувати надійність та ефективність.

4. Відгуки користувачів: Відгуки користувачів безцінні. Перегляньте онлайн-форуми та магазини додатків, щоб дізнатися про реальний досвід, який може виявити потенційні проблеми або видатні функції.

5. Пробні версії: оберіть програмне забезпечення, яке пропонує безкоштовний пробний період. Це дозволить вам безпосередньо випробувати функціональність та оцінити зручність використання без початкових витрат.

6. Підтримка клієнтів: Надійна служба підтримки клієнтів може стати справжнім порятунком у разі технічних проблем. Перевірте, чи пропонує компанія доступні та оперативні варіанти підтримки.

Ретельно досліджуючи та оцінюючи рішення для мобільної безпеки на основі цих факторів, споживачі можуть вибрати продукт, який не лише відповідає їхнім потребам, але й забезпечує душевний спокій.

Розуміння переваг тестування програмного забезпечення для мобільної безпеки

У сучасну цифрову епоху забезпечення безпеки вашого мобільного пристрою є надзвичайно важливим. Але яку користь вам як споживачеві насправді приносить процес тестування та оцінки програмного забезпечення для мобільної безпеки? Давайте розглянемо це детальніше.

Обґрунтоване прийняття рішень

Коли експерти тестують та оцінюють варіанти мобільної безпеки, вони надають вам важливу інформацію про їхню продуктивність та ефективність. Замість того, щоб сліпо вибирати програмне забезпечення, ви маєте доступ до:

– Детальні звіти порівняння: Подивіться, як різні продукти порівнюються один з одним з точки зору функцій та рівнів захисту.

– Показники продуктивності: Зрозумійте, яке програмне забезпечення пропонує найкращий захист без шкоди для продуктивності вашого пристрою.

Посилена гарантія безпеки

Не все програмне забезпечення для мобільної безпеки однакове. Завдяки ретельному тестуванню:

1. Перевірка ефективності: Ви дізнаєтесь, які рішення є найефективнішими для блокування таких загроз, як шкідливе програмне забезпечення, спроби фішингу та підозрілі програми.

2. Оновлений захист: гарантує, що програмне забезпечення може адаптуватися до найновіших загроз безпеці, забезпечуючи безпеку ваших даних.

Ефективність витрат

Інвестування в правильне програмне забезпечення для мобільної безпеки означає розумні витрати. Огляди тестування можуть виділити:

– Співвідношення ціни та якості: Визначте продукти, які пропонують найкращі характеристики за розумною ціною.

– Уникнення помилок: заощаджуйте гроші, уникаючи неякісного програмного забезпечення, яке не забезпечує належного захисту.

Аналітика користувацького досвіду

Окрім безпеки, критично важливим є те, як програма взаємодіє з користувачами. Тестування показує:

– Простота використання: Дізнайтеся, яке програмне забезпечення має зручні інтерфейси, що роблять навігацію легкою.

– Підтримка клієнтів: Зрозумійте рівні підтримки, що пропонуються, якщо у вас виникнуть будь-які проблеми або вам знадобиться допомога.

Перетворюючи комплексне тестування на чіткі, практичні висновки, споживачі краще підготовлені до вибору програмного забезпечення для мобільної безпеки, яке не лише відповідає їхнім потребам, але й забезпечує душевний спокій у цифровому світі.

Для високих вимог безпеки галузі ми повинні забезпечити безпеку передачі даних. Які способи гарантують безпеку даних під час використання GPRS-модемів ? Існує наступний спосіб вирішення проблеми: оператори APN або VPDN впроваджують технологію тунелювання VPDN, що передбачає інкапсуляцію даних у корпоративній мережі для передачі в тунелі. Основний процес тунелювання полягає в тому, що на інтерфейсі між вихідною локальною мережею (LAN) та загальнодоступною мережею дані упаковуються в контейнер у форматі передачі даних загальнодоступної мережі. На інтерфейсі з загальнодоступною мережею LAN цільове рішення інкапсулює дані та знімає навантаження.

Логічний шлях – це інкапсульовані пакети, що передаються через Інтернет, що називається «тунелюванням». Для забезпечення плавної інкапсуляції, передачі та декапсуляції даних використовується протокол зв'язку, який забезпечує безперебійну роботу ядра. Може використовуватися для міжрегіональної групової інтрамережі, спеціалізованої мережі постачальників професійних інформаційних послуг, магістральних мереж, фінансових державних послуг, банківських та інших послуг для доступу до бізнес-мереж. У практичному застосуванні операторам необхідно відкрити відповідний бізнес-

проект, а для його реалізації – прокласти власну лінію зв'язку. Недоліком є громіздкість бізнес-процедур, висока вартість використання та непридатність для загальних проектів. Якщо ви не можете подати заявку на спеціальні мережеві послуги, чи є інші способи забезпечити безпеку даних? Це може бути забезпечено лише апаратним забезпеченням.

У галузі GPRS-модемів зазвичай немає такої функції. У Сямень комунікаційні компанії протягом багатьох років досліджували бездротове термінальне обладнання для передачі даних, і останній продукт компанії – WCTU. Підтримка шифрування даних, підтримка шифрування для DES, 3DES, AES. Під час налаштування WCTU лише запустіть відповідні налаштування. Коли дані WCTU пакетуються, вони використовують шифрування DES, 3DES та AES. Такі дані не турбуються про безпеку в Інтернеті. Ця функція зручна та проста у використанні, її слід популяризувати.

Щоб ефективно керувати своєю мережею та приймати обґрунтовані рішення, вам потрібно збирати інформацію про моделі мережевого трафіку. Якщо у вас виникли проблеми, пов'язані з підключенням або безпекою, вам потрібно мати змогу виявляти зміни в трафіку. Carrier Security надає набір інструментів, що відповідають особливим потребам стільникових мереж.

Журнали та сповіщення відстеження GTP

Служба безпеки оператора записує інформацію про активність сигналізації GTP, що стосується стільникового зв'язку, включаючи APN, IMSI, режим вибору, адреси GSN тощо. Інформація, записана в цих журналах, може допомогти вам визначити, чому певний трафік GTP може бути відхилений або заблокований, а також вирішити, чи слід налаштувати Політику безпеки для прийняття цього трафіку.

Шлюз перевірки GTP Carrier Security генерує широкий спектр детальних сповіщень безпеки у разі порушень протоколу та політики безпеки, включаючи деталі PDU, інформацію про мережу та тип порушення протоколу. Carrier Security також надає сповіщення, специфічні для GTP, про неправильно

сформовані пакети та шкідливу активність.

Запис даних GTP з незрівнянних PDU

Carrier Security може записувати GTP-трафік, який не відповідає правилу GTP у базі правил. Зазвичай трафік, який не відповідає правилу, реєструється в загальному журналі як простий Drop. Carrier Security надає інструмент для збору цих даних за допомогою спеціальних полів, пов'язаних із GTP, які можуть допомогти виявити причину цих падінь.

Бухгалтерський облік GTP

Встановивши правило трафіку користувача GTP на Log, Carrier Security створює запис у журналі для кожного завершеного контексту PDP , який відповідає правилу. У журналі записується загальна кількість користувацьких пакетів (n_pdu) та байтів (n_byte), переданих у площині користувача під час контексту PDP . Carrier Security створює журнали для таких подій:

– Видалення контексту/сеансу PDP.

– Закінчення терміну дії тунелю.

– Відпочинок у тунелі.

– Активний шлюз не працює (у режимі високої доступності).

Захист від надмірних журналів

Через малу кількість пакетів стільникового зв'язку, Carrier Security щодня записує величезну кількість даних, набагато більше, ніж типовий брандмауер Check Point. Цей збір даних є важливим для точної діагностики стану мережі та усунення помилок мережі.

Така інтенсивна активність ведення журналу може зробити деякі системи більш вразливими до атак типу «відмова в обслуговуванні» (DoS). Carrier Security захищає від цього типу атаки, встановлюючи similar loggingпоріг, вище якого аналогічні журнали не генеруються. Цю функцію можна налаштувати.

За замовчуванням – кожні 10 секунд.

Режим лише монітора

Режим лише моніторингу відстежує певний несанкціонований трафік, не

блокуючи його. У цьому режимі брандмауер продовжує перевіряти трафік GTP, але не застосовує жодних засобів захисту, пов'язаних з GTP. Він продовжує застосовувати правила безпеки, пов'язані з GTP, реєструвати активність, пов'язану з GTP, а також створювати журнали помилок і сповіщення GTP. Режим лише моніторингу дозволяє операторам переглядати результати змін глобальних властивостей і налаштувань, що стосуються перевірки GTP. Цей режим корисний для запобігання непередбачуваній поведінці під час першого впровадження безпеки оператора, а також щоразу, коли в глобальні властивості вносяться зміни.

Після ретельного перегляду журналів та переконання, що зміни не перешкоджають легітимному стільниковому трафіку, оператор стільникового зв'язку може вимкнути режим «Тільки моніторинг», а брандмауер може почати блокувати шкідливий GTP-трафік.

Carrier Security слідкує за тунелями GTP та зберігає їх statetак, як це було б у звичайному режимі роботи. Таким чином, ви можете плавно вмикати та вимикати режим лише моніторингу – вся інформація про тунель продовжує існувати в обох режимах, і жодні тунелі не втрачаються під час переходу.

Налаштування моніторингу

– Створювати розширений журнал для незбіганих PDU. Реєструє GTP-пакети, які не збігаються з попередніми правилами, за допомогою розширених полів журналу, пов’язаних з GTP, Carrier Security. Ці журнали мають коричневий колір, а їхній атрибут Action порожній. Значення за замовчуванням – checked.

– Опція відстеження порушення протоколу дозволяє встановити відповідну опцію відстеження або сповіщення, яка використовуватиметься у разі виявлення порушення протоколу (спотвореного пакета). Налаштування за замовчуванням – Log.

3.2 Розробка структурної схеми

GPRS-мережа має зв'язку з великою кількістю зовнішніх мереж (інтернет, роумінг-партнери, корпоративні клієнти, провайдери GRX (GPRS Roaming Exchange), і т.д.). Таке сусідство й партнерські відносини ставлять перед мобільними операторами підвищені вимоги по забезпеченню безпеки переданих даних. Так як зв'язок з партнерами й доступ в інтернет здійснюється по протоколі IP, а усередині GPRS-магістралі дані інкапсулюються в небезпечні тунелі GTP, те границю GPRS-мережі необхідно надійно захищати.

Основну погрозу представляють напрямки Gi і Gp, так як використовуючи протокол IP, будь-який користувач може посилати довільні пакети в GPRS-мережу. Оскільки оператори не обмежують типи користувальницького трафіку, користувачі мобільних терміналів перебувають повністю відкритими для всіх недуг інтернету (віруси, хробаки, трояни, DoS і т.д.). Відповідно, не захищені й користувальницькі дані, які відправляються в зовнішні мережі. Також більшість атак можуть бути спрямовані на саму GPRS-інфраструктуру (Gn), викликаючи відмову або некоректну роботу устаткування.

Типи атак на потенційно небезпечних GPRS-інтерфейсах:

1.Gp-інтерфейс:

– паразитний трафік роумінг-партнера;

– DNS флуд;

– GTP флуд;

– довільне видалення PDP- контекстів користувачів;

– некоректна BGP-інформація;

– підміна DNS-відповідей;

– підміна запитів Create/Update PDP Context;

– overbilling attacks.

2.Gi-інтерфейс:

– DoS-атаки;

– флуд з IP-адрес мобільних станцій;

– влучення в корпоративну мережу іншого клієнта.

3.Gn-інтерфейс:

– підміна GGSN/SGSN;

– підміна запиту видалення PDP-контекстів користувачів;

– атаки мобільних користувачів друг проти друга.

Завдання захисту Gn і Gp інтерфейсів ускладнює особливість комунікацій, які ведуться там по засобом протоколу GTP.

Принцип роботи GTP кардинально відрізняється від інших IP-протоколів: GTP-тунель може бути розподілений по декількох IP-сесіях, і навпаки – одна IP-сесія може містити трохи GTP-тунелів. Такий принцип роботи робить практично марним застосування класичних IP-Firewall для фільтрації GTP-трафіку. Виходячи з того, що звістка IP-трафік на інтерфейсах Gn і Gp інкапсульован в GTP, на даних інтерфейсах необхідний GTP Firewall для перевірки коректності функціонування GTP.

Безліч атак функціонує на більше високих рівнях подання даних (Layer 5-7), ніж ті, з якими працює фаєрвол (Layer 2-4). Тому, як другий ешелон захисту від атак рівня додатків, потрібна система запобігання вторгнень. Аналізуючи легітимний трафік, що пройшов через Firewall на Layer 2-4 дана система проведе високорівневий аналіз і зможе виявити й запобігти атакам, що ховаються в трафіку додатків. Стосовно до GPRS-мережі, такий підхід вибудовування системи захисту особливо ефективний у точці підключення до мережі інтернет (інтерфейс Gi).

Пропонується використовувати ефективне рішення для операторів зв'язку, що здатно забезпечувати безпека трафіку в будь-якій точці GPRS-мережі. Мова йде про лінійку високопродуктивних шлюзів безпеки Juniper High-End SRX з підтримкою GTP. Пристрою Juniper HE SRX – це універсальна модульна платформа, що поєднує функціонала різнорідних пристроїв (firewall, маршрутизатор, комутатор, UTM, IDP) в одному шасі. Всі пристрої працюють під

керуванням надійної платформи JunOS, що успішно зарекомендувала себе в сегменті магістральних рішень. Модульна архітектура SRX дозволяє гнучко масштабувати продуктивність для виконання необхідного функціонала. Підтримка протоколу GTP робить їхнім незамінним рішенням питань безпеки для мобільних GPRS-операторів зв'язку, що легко інтегрується з технологією GPRS Tunneling Protocol.

Впровадження Juniper HE SRX у на найнебезпечніших ділянках GPRS-мережі, а саме, на інтерфейсах Gn і Gp, забезпечить перевірку коректності роботи протоколу GTP і контроль інформаційних потоків між PLMN. А наявність убудованої системи запобігання вторгнень (Juniper IDP) дозволяє High-End SRX забезпечувати фільтрацію трафіку й високоінтелектуальний захист від атак одночасно. Дана особливість оптимально підходить для захисту Gi інтерфейсу GPRS-мережі, що забезпечує взаємодію з небезпечним Інтернет.

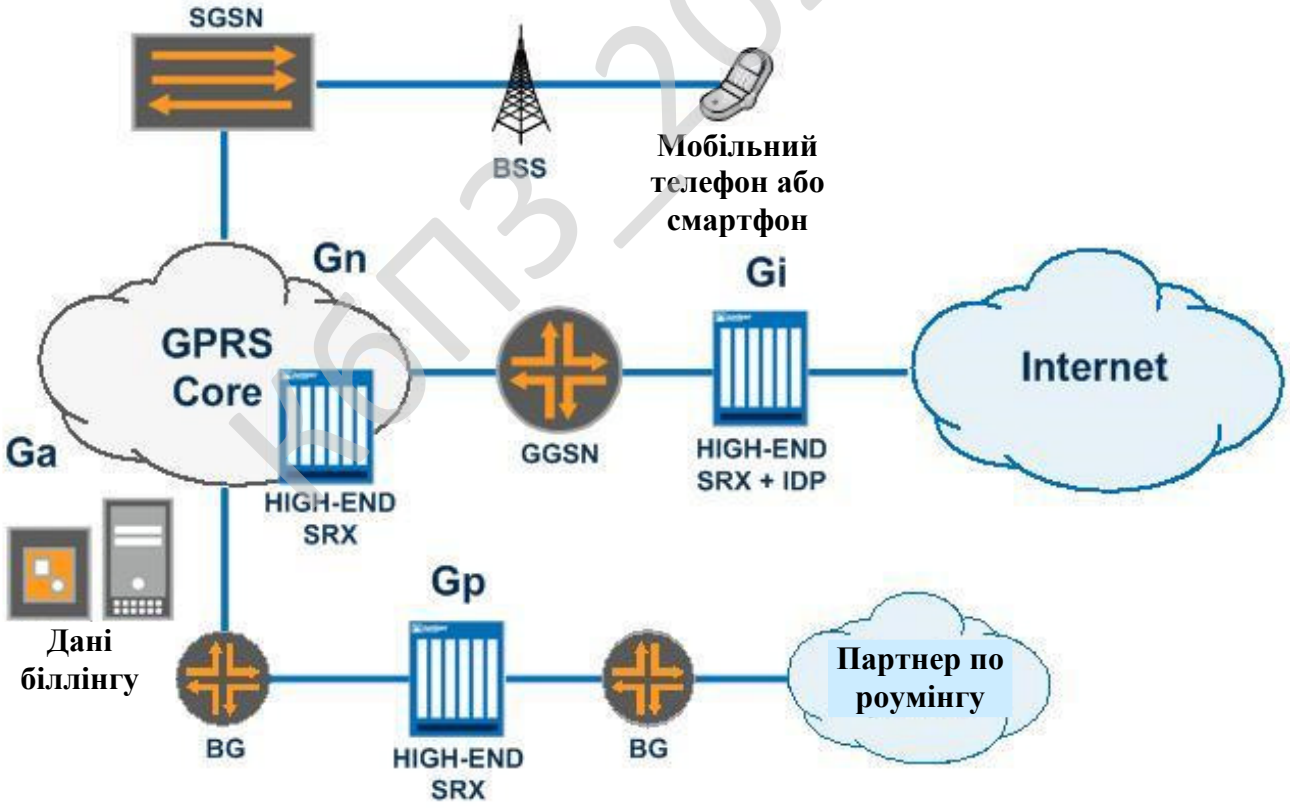

Рисунок 3.1 – Структурна схема системи

При впровадженні SRX важливо враховувати місце в топології GPRS-мережі.

Інтерфейс Gp

У даній точці важливо виділити сервіси, необхідні для взаємодії з роумінг-партнерами прямо або через GRX. У загальному випадку для підключення роумінг-абонента, необхідно забезпечити зв'язок локального SGSN і GGSN його оператора. Для встановлення такого підключення використовується GTP. Крім цього, між PLMN різних операторів повинні функціонувати мережні протоколи BGP і DNS (перетворення імен APN). Основні погрози на Gp інтерфейсі пов'язані з функціонуванням GTP. Для зменшення даних ризиків рекомендовано вживати наступних заходів:

– фільтрація вхідних/вихідних пакетів: запобігає обміну даними з невідомимим роумінг-операторами (при підключенні до GRX) і можливість спуфінг-атак від імені локальної PLMN;

– stateful-фільтрація GTP-пакетів: фільтрація GTP-сесій з невідомими PLMN для запобігання атак і розвантаження локальних GSN;

– обмеження смуги GTP: запобігання DoS-атак, виділення достатньої смуги для роботи GTP, BGP, DNS;

– установлення IPSec-тунелів з роумінгами-партнерами: забезпечення автентифікації й конфіденційності переданих даних;

– запобігання overbilling-атак: повідомлення Gi Firewall про “завислі” сесії для запобігання переплати абонента.

Інтерфейс Gn

Погрози можуть виходити як зсередини мережі оператора, так і бути спрямованими на устаткування мережі. Залежно від інтенсивності атаки можливий варіант тимчасового виводу з ладу устаткування мережі. Це у свою чергу виливається в простої, втрату сервісу, прибутки й невдоволення абонентів. Для усунення даних ризиків рекомендується використовувати політики розмежування доступу й фільтрацію пакетів на основі стану GTP-сесій. Наприклад, у випадку атаки із застосуванням підміни адреси GGSN, запит GTP

PDP Context Delete буде відкинутий якщо перед цим не було GTP PDP Context Create повідомлення.

Інтерфейс Gi

Становить особливу небезпеку, хоч і не вимагає декапсуляції й контролю GTP. Основні механізми захисту включають:

− поділ логічних тунелів для підключення корпоративних клієнтів і впровадження IPSec у випадку використання каналів Інтернет;

− пріоритезація трафіку корпоративних користувачів і IPSec для недопущення можливості відмови каналів абонентів;

− інспектування пакетів з урахуванням стану сесій – використання політик розв'язне тільки ініціювання підключень із боку мобільних станцій;

− фільтрація вхідних/вихідних пакетів – запобігає можливість пересилання даних від IP-адрес мобільної станції, отриманих для виходу в Інтернет, на іншу мобільну станцію;

− запобігання overbilling-атак.

Також загальним підходом при реалізації механізмів безпеки в GPRS-мережах є використання часток IP-адрес для внутрішніх елементів інфраструктури мережі.

Функціонал для безпеки GTP включає:

1. GTP packet sanity check (перевірка заголовка кожного пакета GTP/UDP на відповідність стандарту).

2. GTP stateful inspection (перевірка GTP пакетів на відповідність поточному стану GTP-тунелю в контексті передачі попередніх пакетів; при одержанні пакета не приналежному поточному стану GTP обміну, пристрій відкидає пакет).

3. GGSN redirection (функція перенапрямку запиту GTP PDP Context Create; у запиті вказуються IP-адреси інших GGSN, після чого GTP-U і GTP-C повідомлення посилають зазначеним IP).

4.Policy-based GTP inspection (обмеження доступу між різними PLMN на основі визначення політик безпеки шляхом асоціації PLMN із зонами безпеки).

5. GTP message length filtering (фільтрація GTP-пакетів, що не відповідають мінімальній або максимальній довжині GTP-повідомлення).

6. GTP message type screening (фільтрація GTP-пакетів певного типу).

7. GTP IMSI prefix and APN filtering (фільтрація GTP-пакетів від невідомих PLMN на основі ідентифікатора мережі абонента (IMSI) і шляхи доступу абонента (APN)).

8. Removal of IEs of GTP R6 (функція видалення специфічних атрибутів 3GPP заголовки пакета GTP при наступній передачі в мережі 2GPP).

9. GSN rate limiting (зниження навантаження на GSN за допомогою обмеження швидкості обробки GTP-C пакетів).

10. GTP sequence number validation (функція перевірки порядкових номерів повідомлень G-PDU під час PDP-активації контексту).

11. Cleanup of hanging GTP tunnel (автоматичне видалення “висячих” GTP-тунелів).

12. GTP traffic logging (функція протоколювання GTP-пакетів на основі статусу (forwarded, prohibited, rate-limited, state-invalid, tunnel-limited))

13. GTP tunnel failover for high availability (функція підтримки активних GTP-сесій у режимі відказостійкості).

Переваги:

1. Модульна архітектура.

2. Можливість гнучкого масштабування.

3. Функціонал повноцінного маршрутизатора, фаєрволу, системи запобігання вторгнень і UTM (антивірус, антиспам, веб-фільтр, контент-фильтр).

4. Повна підтримка GTP і механізмів забезпечення його безпеки.

3.3 Розробка функціональної схеми

На рисунку 3.2 представлена функціональна схема системи.

AP – точка доступу.

Процес перевірки виконується в такий спосіб:

– Клієнт формує відповідність GPRS і AP.

– AP посилає клієнтові “ідентифікаційний запит” EAP.

– Клієнт відповідає, посилаючи свої “облікові дані” EAP до AP. Облікові дані складаються з імені користувача й домену.

– AP пересилає даний запит на сервер RADIUS GPRS через неконтрольований порт.

– Після одержання “облікових даних” EAP клієнта сервер RADIUS GPRS запитує сертифікат користувача домену, що відповідає тільки що отриманим обліковим даним, і посилає сертифікат сервера клієнтові.

– AP пересилає запит сертифіката клієнтові.

– Клієнт підтверджує отриманий сертифікат сервера й посилає свій сертифікат користувача домену назад до AP.

– AP пересилає сертифікат серверу RADIUS GPRS.

– Сервер RADIUS GPRS перевіряє його на відповідність за допомогою контролера домену Active Directory і Центра сертифікації, щоб гарантувати, що інформація облікового запису користувача домену в пакеті “облікових даних” відповідає сертифікату користувача домену, отриманому від клієнта. Якщо такої відповідності ні, сервер RADIUS GPRS посилає AP повідомлення про помилку перевірки дійсності.

– При успішній перевірці дійсності сервер RADIUS GPRS посилає AP повідомлення про успішну перевірку дійсності разом із придатним для використання ключем WPA-2 для сеансу.

– AP пересилає ключ WPA-2 клієнтові. Даний ключ WPA-2 використовується протягом усього сеансу, під час якого клієнт зіставлений AP.

– АР відкриває контрольований порт, щоб надати клієнтові доступ до мережних ресурсів.

– Клієнт використовує ключ WPA-2 для шифрування безпечного з'єднання GPRS з АР і запускає DHCP, щоб одержати припустиму IP-адресу.

– Після успішного одержання IP-адреси із сервера DHCP клієнт виконує звичайний вхід у домен і може починати користуватися мережею.

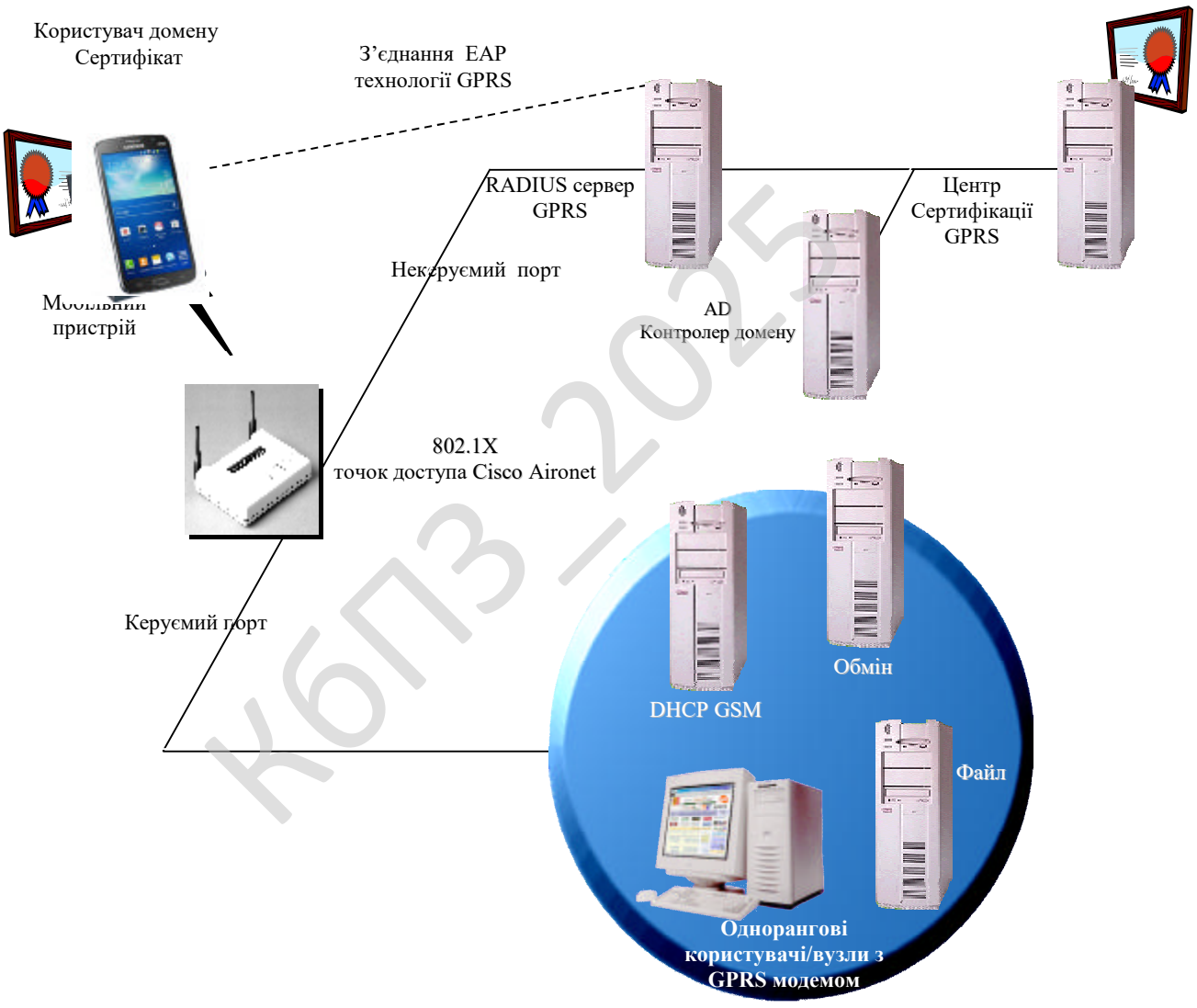

Рисунок 3.2 – Функціональна схема системи

DHCP – Dynamic Host Configuration Protocol – протокол динамічної конфігурації вузла) – це мережний протокол, що дозволяє комп'ютерам

автоматично одержувати IP-адресу й інші параметри, необхідні для роботи в мережі TCP/IP. Даний протокол працює по моделі «клієнт-сервер». Для автоматичної конфігурації комп'ютер-клієнт на етапі конфігурації мережного пристрою звертається до т.зв. сервера DHCP, і одержує від нього потрібні параметри. Мережний адміністратор може задати діапазон адрес, що розподіляються сервером серед комп'ютерів. Це дозволяє уникнути ручного настроювання комп'ютерів мережі й зменшує кількість помилок. Протокол DHCP використовується в більшості великих (і не дуже) мереж TCP/IP.

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

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. Після початку роботи розробленого ПЗ ми потрапляємо до головного блоку системи звідки через ланку дій відбувається наступне:

– Головне вікно ПЗ.

– Автентифікація користувача.

– Створення базового ключа, алгоритм 802.1X.

– Передача ключа точці доступу, протокол TKIP.

– Передача ключа користувачу, протокол TKIP.

– Побудова ієрархії ключів, протокол TKIP.

– Генерація ключів шифрування.

– Шифрування пакетів даних.

– Контроль цілісності, алгоритм MIC.

– Передача даних.

– Дешифрування пакетів даних.

– Перевірка цілісності, алгоритм MIC.

– Завершення сеансу зв’язку.

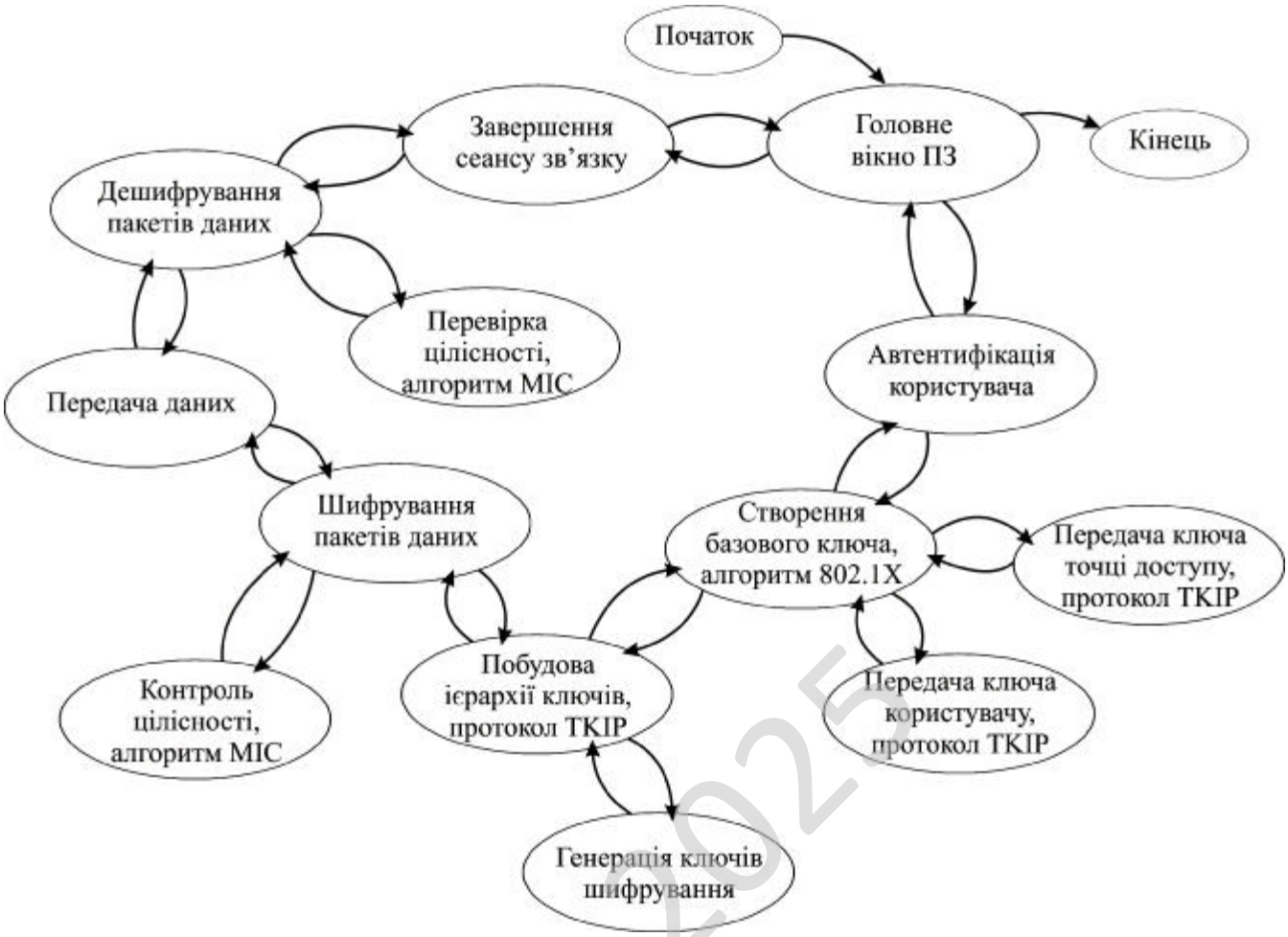

Рисунок 3.3 – Діаграма взаємодії процесів

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі. Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

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

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем. На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З якої видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем я враховував, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об’єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю захисту даних GPRS-мережі.

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою.

Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

Опис алгоритмів функціонування системи

Опис реалізації програмного опитування користувачів. При роботі розробленого програмного продукту в мережі в деяких випадках необхідно знати поточний стан як локального, так і видалених хостів (чи має локальний хост в даний момент можливість виходу в мережу Інтернет, чи доступний якийсь видалений хост і т.д.)

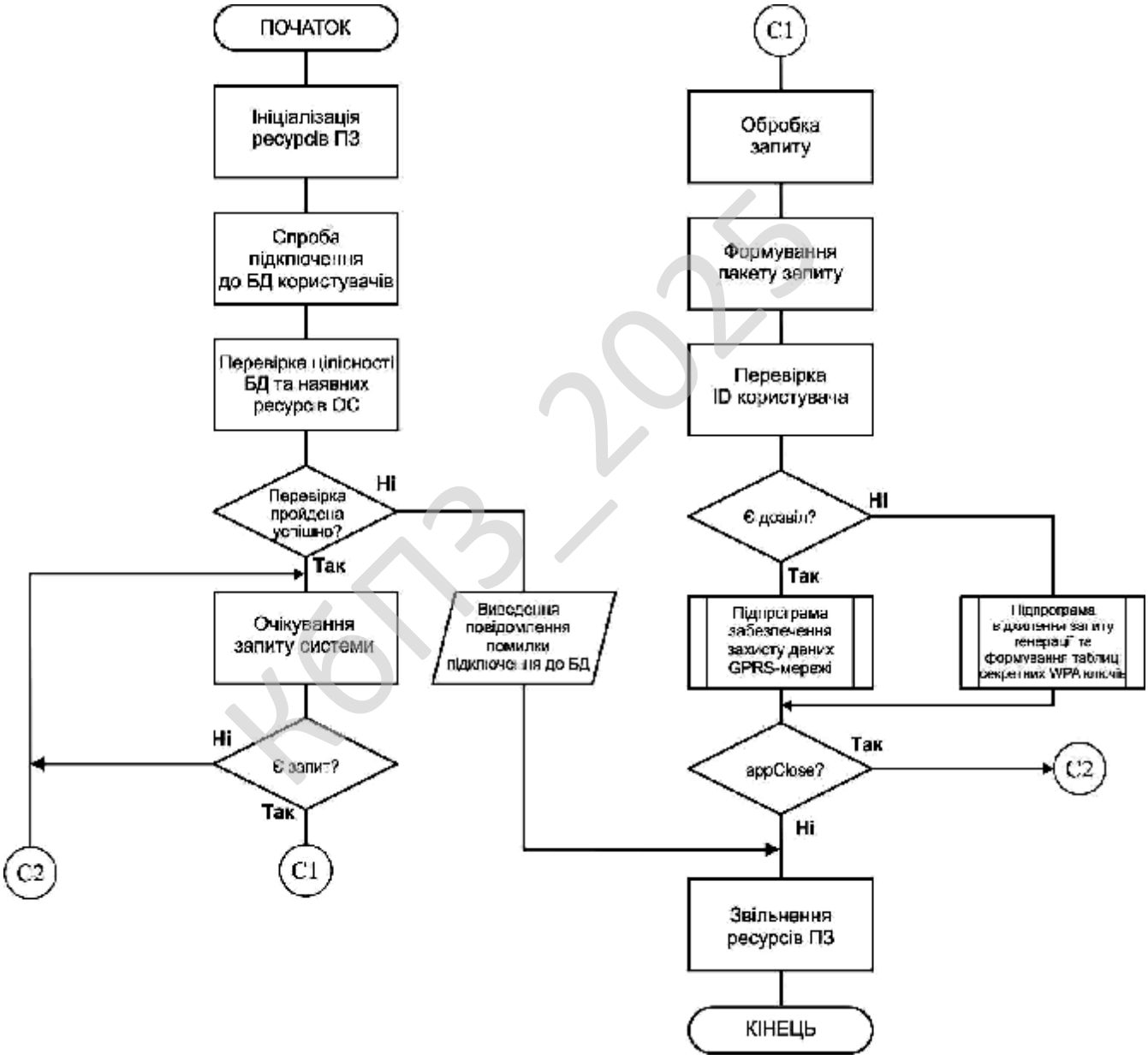

Рисунок 4.1 – Блок схема основної програми

Загальновідомо, що для вказаної мети використовується утиліта ping. Принцип роботи ping-а заснований на використовуванні протоколу ICMP – Internet Control Message Protocol (протокол керівних або контрольних, повідомлень).

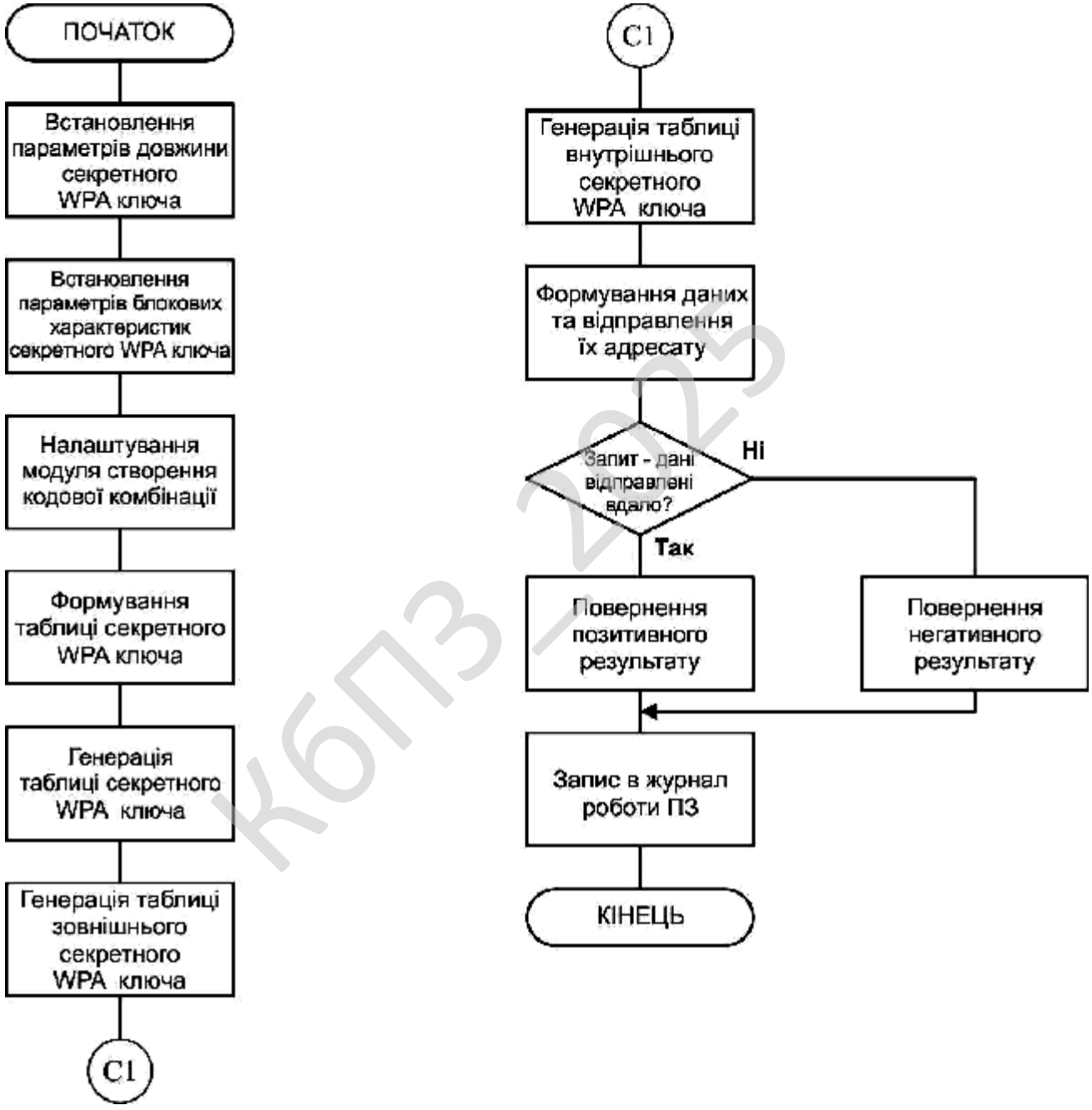

Рисунок 4.2 – Блок схема підпрограми

За допомогою ICMP хост в мережі обмінюються різною службовою інформацією (інформацією про зміну маршруту, зменшення швидкості передачі, неприступність якої-небудь адреси і т.д.)

В основі протоколу ICMP лежить поняття повідомлень. Повідомлення ICMP протоколу, як правило, оповіщають про помилки, що виникають при обробці датаграмм. ICMP використовує основні властивості протоколу IP, неначебто він був протоколом більш високого рівня. На самій же справі ICMP є складовою частиною IP.

Одним з типів повідомлень протоколу є "ехо-запит". Отримавши "ехо-запит" хост зобов'язаний відповісти тому, що послав "ехо-відповіддю".

По суті, "ехо-запит" та "ехо-відповідь" відрізняються лише адресами відправника і одержувача і кодом типу повідомлення (тип 8 – "лунає-запит, тип 0 – "лунає-відповідь).

Реалізації утиліти ping на різних платформах істотно відрізняються. Так, в ОС UNIX використовуються RAW sockets (необроблені, "сирі" сокети), а в ОС Windows всіх версій – ICMP API, Android ICMP.

На практиці, у всіх версіях використовується бібліотека icmp. Отже, на даний момент, можна її використовувати що було зроблено у дипломній роботі.

Опис роботи системи

Система захисту даних GPRS мережі на Android пристроях у цій роботі реалізується як дослідницький програмний прототип на мові Python. Прототип моделює роботу клієнтського програмного агента на Android пристрої, захищеного серверного вузла, а також канал GPRS з типовими для нього затримками, втратою пакетів та потенційним спотворенням трафіку. Такий підхід дає змогу досліджувати поведінку алгоритмів захисту даних без прив’язки до конкретної апаратної конфігурації.

Система працює з абстракцією ідентичності Android пристрою. Ідентичність включає параметри imei, imsi та android_id. Ці атрибути зберігаються у структурі DeviceIdentity. Ідентичність використовується під час

створення сесії, формування криптографічних ключів та під час журналювання подій безпеки.

Основу політики безпеки описує клас SecurityPolicy. У цій політиці задаються мінімальна довжина криптографічного ключа, максимальна тривалість життя сесії, максимальний інтервал неактивності, обмеження на кількість спроб дешифрування та поріг аномальної активності. Політика використовується як на боці сервера, так і на боці клієнта. Це забезпечує узгоджені вимоги до мінімальної стійкості ключів та до правил завершення сесії.

Криптографічні перетворення реалізуються окремим модулем CryptoProvider. Цей модуль працює з майстер ключем та параметрами політики безпеки. Майстер ключ зберігається у змінній master_key та має фіксовану довжину. У реальній Android реалізації майстер ключ розміщується у захищеному сховищі Android Keystore або в апаратному модулі. У прототипі майстер ключ генерується засобами os.urandom та передається до об’єктів клієнта та сервера під час ініціалізації.

Функція derive_session_key у класі CryptoProvider формує сесійний ключ на основі майстер ключа, параметрів ідентичності пристрою та ідентифікатора сесії. Для цього використовується криптографічна геш функція SHA-256. Результат скорочується до довжини, яка задається політикою безпеки. Такий підхід забезпечує унікальність сесійних ключів для різних пристроїв та різних сесій навіть за однакового майстер ключа.

Для забезпечення конфіденційності та цілісності даних застосовується сучасна симетрична криптографія. Функції encrypt_payload та decrypt_payload у модулі CryptoProvider використовують алгоритм AES у режимі GCM на основі бібліотеки cryptography. Режим GCM одночасно забезпечує шифрування та вбудовану перевірку цілісності.

Додаткові дані заголовка кадру передаються як додатково автентифіковані дані. Тому будь яке несанкціоноване змінення заголовка виявляється під час дешифрування.

Додатковий захисний шар реалізується за допомогою обчислення коду автентифікації повідомлення. Для цього функція _compute_hmac обчислює значення HMAC з використанням алгоритму SHA-256. У розрахунок входять нормалізований заголовок кадру та шифротекст у вигляді рядка base64. Результат також кодується у форматі base64.

При прийомі кадру сервер повторює обчислення HMAC і порівнює одержане значення з переданим. Використовується безпечне порівняння з функцією hmac.compare_digest.

Формат переданих повідомлень представляється структурою SecureFrame. Структура має заголовок header, а також рядки nonce, ciphertext та mac. Заголовок містить ідентифікатор сесії, параметри пристрою, ідентифікатор точки доступу apn, мітку часу та тип кадру. Поле nonce зберігає випадковий вектор ініціалізації у форматі base64. Поле ciphertext містить шифротекст у форматі base64. Поле mac містить код автентифікації повідомлення.

Методи to_json та from_json реалізують перетворення між об’єктом SecureFrame та представленням у форматі JSON. Це дозволяє легко зберігати кадри у файлі або передавати їх по мережі.

Для дослідження властивостей GPRS каналу створюється клас SimulatedGprsChannel. Він моделює затримку передачі, ймовірність втрати кадру та ймовірність спотворення шифротексту. При виклику методу transmit система очікує випадкову затримку в заданому діапазоні, а потім з певною ймовірністю або відкидає кадр, або спотворює кілька символів у шифротексті.

Якщо подій втрати та спотворення не відбувається, кадр повертається без змін. Таке моделювання дає змогу перевіряти стійкість алгоритмів до типових для GPRS мережі помилок та затримок.

Серверна частина системи реалізується класом SecureServer. У конструкторі сервер отримує політику безпеки, майстер ключ та посилання на об’єкт журналювання SecurityMonitor.

Сервер зберігає інформацію про активні сесії у словнику sessions. Для

кожної сесії використовується структура SessionState. Вона містить ідентифікатор сесії, ідентичність пристрою, сесійний ключ, час створення сесії, час останньої активності та лічильник невдалих спроб дешифрування.

Процедура створення сесії реалізується методом register_device. Методу передається ідентичність пристрою та пін код. Сервер перевіряє мінімальну довжину пін коду та генерує випадковий ідентифікатор сесії. На основі ідентичності та цього ідентифікатора формується сесійний ключ з використанням CryptoProvider.

Отриманий стан сесії зберігається у словнику. Подія створення сесії фіксується у журналі безпеки методом log_event з типом SESSION_CREATED.

Обробка захищеного кадру виконується методом process_frame. Спочатку сервер отримує заголовок кадру та ідентифікатор сесії. Потім відбувається пошук відповідного запису SessionState.

Сервер перевіряє тривалість життя сесії відповідно до політики. Якщо час перевищує дозволений інтервал, сервер фіксує подію SESSION_EXPIRED, видаляє сесію з словника та не обробляє дані.

Якщо сесія ще є активною, сервер формує словник для дешифрування. У нього входять значення nonce, ciphertext та mac з об’єкта SecureFrame. На основі цих даних та заголовка викликається метод decrypt_payload.

Якщо під час дешифрування або перевірки цілісності виникає виняток, сервер збільшує лічильник спроб у SessionState та фіксує подію DECRYPTION_FAILED із причиною.

Якщо кількість спроб перевищує порогове значення, сервер перевіряє політику та, за потреби, позначає сесію як заблоковану подією SESSION_BLOCKED та видаляє стан сесії.

У разі правильного дешифрування сервер реєструє успішну обробку кадру у журналі подією DATA_FRAME_ACCEPTED. Далі сервер аналізує тип кадру. Якщо тип дорівнює MEASUREMENT, сервер формує відповідь із статусом OK, передає отримані від клієнта дані та додає серверну мітку часу. Якщо кадр

містить запит політики, сервер повертає скорочене представлення параметрів політики у вигляді словника. Це дає змогу клієнту динамічно адаптувати свою поведінку до вимог безпеки.

Клієнтська частина системи моделюється класом AndroidDeviceAgent. У конструкторі агент отримує ідентичність пристрою, параметри точки доступу apn, політику безпеки, майстер ключ, посилання на сервер та посилання на канал SimulatedGprsChannel.

Агент створює власний екземпляр CryptoProvider з тим самим майстер ключем. Це дає змогу незалежно обчислювати сесійні ключі на стороні клієнта.

Початкове встановлення захищеної сесії реалізується методом perform_handshake. Агент викликає на сервері метод register_device, передає ідентичність та пін код та отримує ідентифікатор сесії. Потім клієнт викликає CryptoProvider.derive_session_key з власною ідентичністю та отриманим ідентифікатором.

Таким чином клієнт та сервер незалежно одержують однаковий сесійний ключ без його передачі по мережі. У реальній GPRS мережі такий механізм доповнюється процедурою розподілу майстер ключа і захищеною реалізацією на Android пристрої.

Для передачі вимірювальних даних використовується метод send_measurement. Спочатку агент формує заголовок кадру методом _build_header. У заголовку містяться ідентифікатор сесії, параметры ідентичності, apn, мітка часу та тип кадру. Потім метод encrypt_payload шифрує корисні дані вимірювань та формує криптографічну обгортку.

На основі заголовка та криптографічних полів агент створює об’єкт SecureFrame. Цей об’єкт передається до каналу SimulatedGprsChannel. Канал або повертає кадр без змін, або моделює втрату чи спотворення.

Після цього агент передає доставлений кадр у метод process_frame на сервері та отримує відповідь у вигляді словника.

Запит політики безпеки реалізується методом request_policy. Механізм

формування кадру аналогічний процесу відправки вимірювань, однак тип кадру встановлюється як POLICY_REQUEST, а корисні дані містять службовий параметр request. Сервер повертає скорочене представлення політики, яке клієнт може використати для аналізу дозволених режимів шифрування, тривалості сесії та інших налаштувань.

Журналювання та моніторинг безпеки реалізуються класом SecurityMonitor. Клас зберігає події у файлі формату JSON Lines. Функція log_event записує об’єкти з полями time, type та details. Для оперативного моніторингу одночасно використовується стандартний модуль logging. Метод detect_anomalies аналізує журнал та шукає часові вікна з надмірною кількістю подій DATA_FRAME_ACCEPTED. Такий підхід дає змогу виявляти можливі спроби автоматизованого перебору або нетипову активність окремих пристроїв.

Розглянемо вихідний код з описом основних функцій.

```
import logging
import random
import hashlib
import uuid
from dataclasses import dataclass, field
from typing import Dict, Any, List, Optional

from cryptography.hazmat.primitives.ciphers.aead import AESGCM

# Модуль демонструє програмну реалізацію системи захисту даних GPRS мережі для
Android пристроїв
# Код містить серверну частину, клієнтський агент, симульований канал GPRS та
підсистему моніторингу безпеки

GPRS_DEFAULT_APN = "internet.operator.example"
GPRS_MAX_FRAME_SIZE = 1400
SESSION_ID_BYTES = 16
NONCE_LENGTH = 12
MASTER_KEY_LENGTH = 32
LOG_FILE_NAME = "gprs_security_log.jsonl"
```

```
# Клас SecurityPolicy зберігає параметри політики безпеки для захисту даних у GPRS мережі
@dataclass
class SecurityPolicy:
    min_key_length: int = 32
    max_session_lifetime: int = 900
    max_inactive_interval: int = 300
    max_retries: int = 3
    allowed_ciphers: List[str] = field(default_factory=lambda: ["AESGCM"])
    anomaly_threshold_per_minute: int = 60

# Клас DeviceIdentity описує ідентичність Android пристрою в GPRS мережі
@dataclass
class DeviceIdentity:
    imei: str
    imsi: str
    android_id: str

    def as_dict(self) -> Dict[str, str]:
        return {
            "imei": self.imei,
            "imsi": self.imsi,
            "android_id": self.android_id,
        }

# Клас SessionState зберігає стан захищеної сесії між пристроєм та сервером
@dataclass
class SessionState:
    session_id: str
    device: DeviceIdentity
    session_key: bytes
    created_at: float
    last_seen: float
    retry_counter: int = 0

# Клас SecureFrame представляє захищений кадр, який передається каналом GPRS
@dataclass
class SecureFrame:
    header: Dict[str, Any]
    nonce: str
    ciphertext: str
```

```
    mac: str

    def to_json(self) -> str:
        return json.dumps(
            {
                "header": self.header,
                "nonce": self.nonce,
                "ciphertext": self.ciphertext,
                "mac": self.mac,
            },
            ensure_ascii=False,
        )

    @staticmethod
    def from_json(data: str) -> "SecureFrame":
        raw = json.loads(data)
        return SecureFrame(
            header=raw["header"],
            nonce=raw["nonce"],
            ciphertext=raw["ciphertext"],
            mac=raw["mac"],
        )

# Клас CryptoProvider відповідає за всі криптографічні операції у системі
class CryptoProvider:
    def __init__(self, master_key: bytes, policy: SecurityPolicy):
        self.master_key = master_key
        self.policy = policy

    def derive_session_key(self, identity: DeviceIdentity, session_id: str) ->
bytes:
        material = (
            identity.imsi
            + identity.imei
            + identity.android_id
            + session_id
        ).encode("utf-8")
        digest = hashlib.sha256(material + self.master_key).digest()
        key = digest[: self.policy.min_key_length]

   return key
```

```
    def _compute_hmac(self, header: Dict[str, Any], ciphertext_b64: str) -> str:
        message = json.dumps(header, sort_keys=True).encode("utf-8") +
ciphertext_b64.encode("ascii")
        tag = hmac.new(self.master_key, message, hashlib.sha256).digest()
        return base64.b64encode(tag).decode("ascii")

    def encrypt_payload(self, session_key: bytes, header: Dict[str, Any], payload:
Dict[str, Any]) -> Dict[str, str]:
        aesgcm = AESGCM(session_key)
        nonce = os.urandom(NONCE_LENGTH)
        plaintext = json.dumps(payload, ensure_ascii=False).encode("utf-8")
        aad = json.dumps(header, sort_keys=True).encode("utf-8")
        ciphertext = aesgcm.encrypt(nonce, plaintext, aad)
        ciphertext_b64 = base64.b64encode(ciphertext).decode("ascii")
        nonce_b64 = base64.b64encode(nonce).decode("ascii")
        mac = self._compute_hmac(header, ciphertext_b64)
        return {
            "nonce": nonce_b64,
            "ciphertext": ciphertext_b64,
            "mac": mac,
        }

    def decrypt_payload(self, session_key: bytes, header: Dict[str, Any],
frame_body: Dict[str, str]) -> Dict[str, Any]:
        expected_mac = self._compute_hmac(header, frame_body["ciphertext"])
        received_mac = frame_body["mac"]
        if not hmac.compare_digest(expected_mac, received_mac):
            raise ValueError("Помилка автентифікації кадру")
        nonce = base64.b64decode(frame_body["nonce"])
        ciphertext = base64.b64decode(frame_body["ciphertext"])
        aesgcm = AESGCM(session_key)
        aad = json.dumps(header, sort_keys=True).encode("utf-8")
        plaintext = aesgcm.decrypt(nonce, ciphertext, aad)
        data = json.loads(plaintext.decode("utf-8"))
        return data

# Клас SimulatedGprsChannel моделює властивості GPRS каналу з втратою і
спотворенням кадрів
class SimulatedGprsChannel:
    def __init__(
        self,
```

```
        loss_probability: float = 0.01,
        corruption_probability: float = 0.005,
        delay_range: Optional[tuple] = None,
    ):
        self.loss_probability = loss_probability
        self.corruption_probability = corruption_probability
        if delay_range is None:
            self.delay_range = (0.1, 0.5)
        else:
            self.delay_range = delay_range

    def transmit(self, frame: SecureFrame) -> Optional[SecureFrame]:
        delay = random.uniform(self.delay_range[0], self.delay_range[1])
        time.sleep(delay)
        if random.random() < self.loss_probability:
            return None
        if random.random() < self.corruption_probability:
            corrupted = SecureFrame(
                header=dict(frame.header),
                nonce=frame.nonce,
                ciphertext=frame.ciphertext,
                mac=frame.mac,
            )
            if corrupted.ciphertext:
                index = random.randint(0, len(corrupted.ciphertext) - 1)
                replacement = "A"
                corrupted.ciphertext = (
                    corrupted.ciphertext[: index]
                    + replacement
                    + corrupted.ciphertext[index + 1 :]
                )
            return corrupted
        return frame

# Клас SecurityMonitor відповідає за журналювання подій та базовий аналіз аномалій
class SecurityMonitor:
    def __init__(self, log_file: str = LOG_FILE_NAME):
        self.log_file = log_file
        logging.basicConfig(
            level=logging.INFO,
            format="%(asctime)s %(levelname)s %(message)s",
        )
```

```
        self.logger = logging.getLogger("gprs_security")

    def log_event(self, event_type: str, details: Dict[str, Any]) -> None:
        record = {
            "time": time.time(),
            "type": event_type,
            "details": details,
        }
        with open(self.log_file, "a", encoding="utf-8") as fh:
            fh.write(json.dumps(record, ensure_ascii=False) + "\n")
        self.logger.info("Подія %s зафіксована", event_type)

    def detect_anomalies(self) -> List[Dict[str, Any]]:
        try:
            with open(self.log_file, "r", encoding="utf-8") as fh:
                records = [json.loads(line) for line in fh if line.strip()]
        except FileNotFoundError:
            return []
        if not records:
            return []
        records.sort(key=lambda r: r["time"])
        window = 60.0
        anomalies: List[Dict[str, Any]] = []
        policy = SecurityPolicy()
        start_time = records[0]["time"]
        current_window_start = start_time
        last_time = records[-1]["time"]
        while current_window_start <= last_time:
            window_end = current_window_start + window
            count = sum(
                1
                for r in records
                if current_window_start <= r["time"] < window_end
                and r["type"] == "DATA_FRAME_ACCEPTED"
            )
            if count > policy.anomaly_threshold_per_minute:
                anomalies.append(
                    {
                        "window_start": current_window_start,
                        "window_end": window_end,
                        "count": count,
                    }
```

```
                )
            current_window_start += window
        return anomalies

# Клас SecureServer реалізує серверну частину захищеної GPRS системи
class SecureServer:
    def __init__(self, policy: SecurityPolicy, master_key: bytes, monitor:
Optional[SecurityMonitor] = None):
        self.policy = policy
        self.crypto = CryptoProvider(master_key, policy)
        if monitor is None:
            self.monitor = SecurityMonitor(LOG_FILE_NAME)
        else:
            self.monitor = monitor
        self.sessions: Dict[str, SessionState] = {}

    def register_device(self, identity: DeviceIdentity, pin: str) -> str:
        if not pin or len(pin) < 4:
            raise ValueError("Пін код має недостатню довжину")
        session_id = uuid.uuid4().hex[: SESSION_ID_BYTES * 2]
        session_key = self.crypto.derive_session_key(identity, session_id)
        state = SessionState(
            session_id=session_id,
            device=identity,
            session_key=session_key,
            created_at=time.time(),
            last_seen=time.time(),
        )
        self.sessions[session_id] = state
        self.monitor.log_event(
            "SESSION_CREATED",
            {
                "session_id": session_id,
                "imei": identity.imei,
                "imsi": identity.imsi,
            },
        )
        return session_id

    def _get_session(self, session_id: str) -> SessionState:
        if session_id not in self.sessions:
```

```
            raise KeyError("Сесія відсутня на сервері")
        return self.sessions[session_id]

    def _export_policy(self) -> Dict[str, Any]:
        return {
            "min_key_length": self.policy.min_key_length,
            "max_session_lifetime": self.policy.max_session_lifetime,
            "max_inactive_interval": self.policy.max_inactive_interval,
            "max_retries": self.policy.max_retries,
        }

    def process_frame(self, frame: SecureFrame) -> Optional[Dict[str, Any]]:
        header = frame.header
        session_id = header.get("session_id")
        if not session_id:
            self.monitor.log_event(
                "FRAME_REJECTED",
                {
                    "reason": "відсутній ідентифікатор сесії",
                },
            )
            return None
        try:
            state = self._get_session(session_id)
        except KeyError:
            self.monitor.log_event(
                "FRAME_REJECTED",
                {
                    "reason": "невідома сесія",
                    "session_id": session_id,
                },
            )
            return None
        now = time.time()
        if now - state.created_at > self.policy.max_session_lifetime:
            self.monitor.log_event(
                "SESSION_EXPIRED",
                {
                    "session_id": session_id,
                    "imei": state.device.imei,
                },
            )
```

```
            del self.sessions[session_id]
            return None
        state.last_seen = now
        frame_body = {
            "nonce": frame.nonce,
            "ciphertext": frame.ciphertext,
            "mac": frame.mac,
        }
        try:
            data = self.crypto.decrypt_payload(state.session_key, header,
frame_body)
        except Exception as exc:
            state.retry_counter += 1
            self.monitor.log_event(
                "DECRYPTION_FAILED",
                {
                    "session_id": session_id,
                    "error": str(exc),
                    "retries": state.retry_counter,
                },
            )
            if state.retry_counter >= self.policy.max_retries:
                self.monitor.log_event(
                    "SESSION_BLOCKED",
                    {
                        "session_id": session_id,
                        "imei": state.device.imei,
                    },
                )
                del self.sessions[session_id]
            return None
        self.monitor.log_event(
            "DATA_FRAME_ACCEPTED",
            {
                "session_id": session_id,
                "imei": state.device.imei,
                "payload_size": len(json.dumps(data, ensure_ascii=False)),
                "timestamp": header.get("timestamp"),
                "type": header.get("type"),
            },
        )
        frame_type = header.get("type")
```

```
        if frame_type == "MEASUREMENT":
            return {
                "status": "OK",
                "received": data,
                "server_time": now,
            }
        if frame_type == "POLICY_REQUEST":
            return {
                "status": "OK",
                "policy": self._export_policy(),
            }
        return {
            "status": "UNKNOWN_TYPE",
            "server_time": now,
        }
```

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою CRYPTON – алгоритм симетричного блочного шифрування (розмір блоку 128 біт, ключ довжиною до 256 біт), розроблений південнокорейським криптологом Чьо Лім Хун з південнокорейської компанії Future Systems, яка з кінця 1980-х років працює на ринку забезпечення мереж і захисту інформації. Алгоритм був розроблений в 1998 році в якості шифру – учасника конкурсу AES.

Як зізнавався автор, конструкція алгоритму спирається на алгоритм SQUARE[1]. В алгоритмі Crypton немає традиційних для блочних шифрів мережі Фейстеля.

Основу даного шифру становить так звана SP-мережа (повторювана циклова функція, що складається із замін-перестановок, орієнтована на розпаралелену нелінійну обробку всього блоку даних). Крім високої швидкості, перевагами таких алгоритмів є полегшення дослідження стійкості шифру до методів диференціального та лінійного криптоаналізу, що є на сьогодні основними інструментами розтину блочних шифрів.

На конкурс AES була представлена версія алгоритму Crypton v0.5. Однак, як казав Чьо Лім Хун, йому не вистачало часу для розробки повної версії. І вже на першому етапі конкурсу AES в ході аналізу алгоритмів, версія Crypton v0.5 була замінена на версію Crypton v1.0. Відмінність нової версії від первинної полягала в зміні таблиці замін та в модифікації процесу розширення ключа.

Як і інші учасники конкурсу AES, Crypton призначений для шифрування 128-бітових блоків даних[2]. При шифруванні використовуються ключі шифрування для декількох фіксованих розмірів – від 0 до 256 біт з кратністю 8 бітів.

Структура алгоритму Crypton – структура «Квадрата» – багато в чому схожа на структуру алгоритму Square, створеного в 1997 році. Криптографічні перетворення для алгоритмів з даною структурою можуть бути виконані як для цілих рядків і стовпців масиву, так і над окремими його байтами. (Варто зазначити, що алгоритм Square був розроблений авторами майбутнього переможця конкурсу AES – авторами алгоритму Rijndael – Вінсентом Ріджменом і Джоан Дейменом.)

Шифрування

Алгоритм Crypton являє 128-бітовий блок шифруємих даних у вигляді байтового масиву 4×4, над якими в процесі шифрування проводиться кілька раундів перетворень. У кожному раунді передбачається послідовне виконання наступних операцій:

– Таблична заміна γ;

– Лінійне перетворення π;

– Байтова перестановка τ;

– Операція σ.

Таблична заміна γ

Алгоритм Crypton використовує 4 таблиці замін. Кожна з яких заміщає 8-бітне вхідне значення на вихідне такого ж розміру.

Лінійне перетворення π

Тут використовується 4 спеціальні константи. Ці константи об'єднані в маскуючі послідовності

Байтова перестановка τ

Дана перестановка перетворює найпростішим чином рядок даних у стовпець.

Операція σ

Дана операція є побітовим складанням всього масиву даних з ключем раунду. Зауважимо, саме 12 раундів шифрування рекомендується автором алгоритму Чьо Хун Лімом, проте сувора кількість раундів не встановлена.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна програмного продукту розподілено на наступні розділи:

– Алгоритм шифрування.

– Початковий спільний ключ WPA-2.

– Тимчасовий проміжок зміни ключів.

– IP адреса RADIUS сервера.

– Порт RADIUS сервера.

Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

Рисунок 5.1 – Основне вікно програми

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

Рисунок 5.2 – Авторське право

Обрано умови розповсюдження – proprietary software.

Програмне забезпечення, на яке зберігаються як немайнові, так і майнові авторські права. Отримавши або придбавши таке програмне забезпечення, користувач отримує обмежені права користування ним: може бути заборонено або закрито доступ до коду (вивчення), внесення змін, тиражування, розповсюдження та перепродаж. Програмне забезпечення вважається власницьким, якщо наявне хоча б одне з перелічених обмежень.

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

Метою розробки є дослідження та програмна реалізація системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

Об'єктом дослідження є процес захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

Предметом дослідження є методи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

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

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

– Удосконалено метод захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

– Розроблено вітчизняний продукт захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android, який має більш широкі можливості, на відміну від існуючих аналогів.

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати цього дослідження можуть бути цікавими передусім компаніям, що активно використовують мобільні пристрої у своїй роботі – наприклад, підприємствам логістики, сервісним службам, банківським структурам, а також державним установам, які збирають та обробляють конфіденційну інформацію поза офісом. У таких організаціях передача даних через GPRS-з'єднання є критично важливою частиною бізнес-процесів, а ризики їх перехоплення або втрати можуть спричинити значні фінансові та репутаційні збитки.

Також система може становити інтерес для розробників корпоративних мобільних додатків, які прагнуть інтегрувати в свої продукти надійні інструменти безпеки. Вона дозволить їм запропонувати клієнтам більш стійкі до атак рішення, що відповідатимуть сучасним вимогам безпеки мобільного середовища.

Не менш важливими потенційними користувачами є оператори зв'язку та інтегратори, які забезпечують підключення мобільних пристроїв до корпоративних мереж. Для них програмна реалізація такого типу відкриває можливості надання додаткових послуг – наприклад, безпечного тунелювання трафіку, управління ключами шифрування або моніторингу активності користувачів у режимі реального часу.

Крім того, результати дослідження можуть бути корисними для освітніх та наукових установ, які займаються кібербезпекою або дослідженнями в галузі мобільних технологій. Вони можуть використовувати цю систему як приклад практичного впровадження алгоритмів шифрування, автентифікації та управління доступом у середовищі Android.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості системи може бути проведена методом експертного аналізу, залучаючи фахівців із кібербезпеки, розробників мобільних додатків, системних адміністраторів та представників бізнесу, які працюють із мобільними технологіями. Кожен експерт оцінює систему за визначеними критеріями – рівнем інноваційності, масштабованості, простотою впровадження, рентабельністю, потребою ринку та рівнем безпеки.

Наприклад, десять експертів можуть виставити оцінки за шкалою від 1 до 10. Середній бал за інноваційність – 8,9, за простоту інтеграції – 7,5, за потенційну економічну вигоду – 9,2, за затребуваність на ринку – 8,8, а за безпечність – 9,7. Після нормалізації цих значень середній інтегральний показник привабливості становитиме близько 8,8 бала, що свідчить про високий рівень перспективності продукту.

Цей підхід дозволяє не лише кількісно оцінити доцільність впровадження, але й зрозуміти, які аспекти потребують доопрацювання. Якщо, наприклад, експерти зазначать, що інтеграція із вже наявними корпоративними системами може бути складною, це сигнал до оптимізації архітектури або створення додаткових API.

Таким чином, метод експертних оцінок не лише підтверджує привабливість проєкту, а й виступає ефективним інструментом для визначення слабких місць, що підвищує якість кінцевого продукту.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості доцільно застосувати метод функціонально-вартісного аналізу (FVA) у поєднанні з методом життєвого циклу продукту (Life-Cycle Costing). Такий підхід дозволяє врахувати не лише початкові витрати на

розробку та впровадження системи, але й витрати на її обслуговування, оновлення, навчання користувачів і модернізацію у майбутньому.

FVA допоможе розділити систему на функціональні блоки – модуль шифрування трафіку, автентифікацію користувачів, систему моніторингу загроз тощо – і визначити їхній внесок у загальну вартість. Це дозволяє оптимізувати бюджет, концентруючись на тих компонентах, які дають найбільшу користь при мінімальних витратах.

Метод життєвого циклу, своєю чергою, дозволяє оцінити витрати на етапах розробки, впровадження, підтримки й утилізації. Наприклад, інвестиції у перший рік можуть становити 450 000 грн, але завдяки зниженню кількості інцидентів система приносить економію понад 2 млн грн щорічно. Таким чином, окупність досягається вже за кілька місяців.

Поєднання цих методів дозволяє створити реалістичну фінансову модель, яка враховує не лише короткострокові вигоди, а й довгострокову стійкість системи. Це важливо, оскільки захист даних – це не одноразовий захід, а постійний процес, який має залишатися економічно виправданим у перспективі.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Економічна ефективність від впровадження системи захисту даних GPRS-мережі, розгорнутої на пристроях під операційною системою Android розраховується наступним чином. Вхідні дані зафіксовано в таблиці 7.1.

Розрахунок економічного ефекту демонструє наступне: зменшення кількості інцидентів – 480 000грн, скорочення витрат на підтримку – 520 000грн, підвищення ефективності працівників 75 000грн, подовження життєвого циклу пристроїв – 1 000 000грн, загальний річний ефект – 2 075 000 грн, мінус щорічна підтримка системи -–100 000 грн, чистий річний ефект -1 975 000 грн, термін окупності – 0,23 року (~3 місяці), ROI (Return on Investment) ≈ 438%.

Таблиця 7.1 – Вихідні дані для розрахунку

Показник	**До впровадження**	**Після впровадження**
Кількість інцидентів безпеки на рік	5	1
Середня вартість ліквідації інциденту	120 000 грн	40 000 грн
Рівень втрат даних (витік/втрата пристрою)	4%	0,5%
Витрати на підтримку (усунення наслідків атак)	600 000 грн/рік	80 000 грн/рік
Продуктивність працівників (час простою системи)	−3% від річного часу	−0,5%
Тривалість життя пристроїв (зменшення відновлень системи)	2 роки	2,5 року
Вартість впровадження системи	—	450 000 грн (одноразово)
Щорічне обслуговування системи	—	100 000 грн

Додаткові (непрямі) вигоди: зниження репутаційних ризиків: відсутність витоків даних зменшує загрозу втрати клієнтів і штрафів за порушення GDPR, відповідність стандартам безпеки – ISO 27001, PCI DSS, Android Enterprise Recommended, спрощення адміністрування – централізоване оновлення політик безпеки і швидке відновлення пристроїв після інцидентів, аналітика та прогнозування загроз – система збирає статистику атак, що дозволяє формувати профілактичні заходи, підвищення довіри клієнтів і партнерів, особливо у сферах банківських, логістичних і страхових послуг.

У довгостроковій перспективі система створює фундамент для подальшої цифрової безпеки компанії, дозволяючи інтегруватися з VPN-сервером,

корпоративними SIEM-платформами і системами поведінкового аналізу. Таким чином, інвестиція в захист GPRS-комунікацій перетворюється не лише на засіб економії, а й на стратегічний елемент стійкості бізнесу.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Першим етапом просування має бути підготовка демонстраційного середовища, у якому потенційні клієнти можуть на практиці побачити, як система працює. Для цього можна створити тестову мережу з імітацією GPRS-з'єднань і показати реальні сценарії – наприклад, як система реагує на спробу перехоплення трафіку або несанкціонований доступ. Це створить довіру до продукту.

Другим кроком стане позиціювання рішення через професійні конференції, форуми з кібербезпеки, а також через публікації у спеціалізованих ЗМІ. Участь у таких заходах дозволяє залучити не лише потенційних клієнтів, а й партнерів – інтеграторів, операторів мобільного зв'язку та виробників пристроїв.

Третім етапом може бути співпраця з державними структурами або великими корпораціями у форматі пілотних проєктів. Реальні кейси успішного впровадження значно підвищують довіру ринку. Після цього доцільно запустити SaaS-модель – систему як сервіс, доступний за підпискою, що зменшить бар'єр входу для нових користувачів.

Таким чином, просування має поєднувати технічні демонстрації, експертну присутність у професійному середовищі та гнучку модель комерціалізації, яка робить продукт доступним для різних категорій клієнтів.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації збуту важливо створити мультиканальну стратегію. Основний акцент варто зробити на співпраці з мобільними операторами, які можуть пропонувати систему як частину своїх корпоративних тарифів або послуг

для бізнесу. Це дозволить відразу охопити велику кількість користувачів без необхідності індивідуального залучення кожного клієнта.

Паралельно можна розробити онлайн-платформу, де користувачі матимуть змогу самостійно протестувати систему у хмарному середовищі, оформити підписку або ліцензію. Такий підхід значно спрощує доступ до продукту і робить процес продажу майже повністю автоматизованим.

Ще одним ефективним шляхом є співпраця з компаніями, які розробляють корпоративні мобільні рішення, зокрема CRM або логістичні системи. Інтеграція модуля захисту як додаткової функції в їхні продукти дозволить розширити аудиторію без прямих витрат на маркетинг.

Оптимізація збуту також можлива через навчальні курси з мобільної безпеки, де система може використовуватись як приклад практичного інструменту. Це не лише підвищить впізнаваність продукту, але й створить новий канал залучення користувачів через освітнє середовище.

7.7 Визначення ключових факторів успіху конкретного проєкту

Головним фактором успіху є ефективність системи у реальних умовах – наскільки вона справді здатна запобігти втраті або викраденню даних без негативного впливу на швидкість роботи пристрою. Якщо користувач не помічає уповільнення або складнощів при використанні, це створює позитивний користувацький досвід і стимулює подальше впровадження.

Другим важливим чинником є масштабованість і гнучкість рішення. Система повинна легко адаптуватися під різні конфігурації Android-пристроїв і типи мереж. Це дозволить залучати клієнтів із різних секторів – від малого бізнесу до державних структур.

Не менш значущим є рівень підтримки клієнтів. Якщо розробники забезпечують оперативне реагування на технічні проблеми, оновлення та захист

від нових типів загроз, система поступово набуває репутації надійного і стабільного продукту.

І, зрештою, успіх визначається довірою ринку. Система, що демонструє реальні результати – скорочення кількості інцидентів, підвищення безпеки і зниження витрат, – стає не просто технічним рішенням, а стратегічним активом для бізнесу, який допомагає забезпечити його стійкість у цифровому середовищі.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Згідно із Законом України про охорону праці, маємо визначення: "Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі праці", тож в рамках цього визначення маємо, що техніка безпеки – це система правових, соціально-економічних, організаційно-технічних та санітарно-гігієнічних заходів та засобів, спрямованих на збереження життя та здоров'я людини під час її трудової діяльності, тобто це низка заходів по створенню безпечних умов виробництва для працівників під час виконання ними своїх трудових обов'язків, а отже, відповідно державна політика України спрямована на створення належних, безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням. Основою державної політики в цьому питанні є:

– пріоритет життя і здоров'я працівників, повна відповідальність роботодавця за створення належних, безпечних і здорових умов праці;

– підвищення рівня промислової безпеки шляхом забезпечення суцільного технічного контролю за станом виробництва, технологій та продукції, а також сприяння підприємствам у створенні безпечних та нешкідливих умов праці;

– комплексне розв'язання завдань охорони праці на основі загальнодержавної, галузевих, регіональних програм з цього питання та з урахуванням інших напрямів економічної і соціальної політики, досягнень в галузі науки і техніки та охорони довкілля;

– соціальний захист працівників (повне відшкодування шкоди особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань);

– встановлення єдиних вимог з охорони праці для всіх підприємств та суб'єктів підприємницької діяльності незалежно від форм власності та видів діяльності;

– адаптація трудових процесів до можливостей працівника з урахуванням його здоров'я та психологічного стану;

– пріоритет у використанні економічних методів управління охороною праці (участі держави у фінансуванні заходів щодо охорони праці, залучення добровільних внесків та інших надходжень на ці цілі, отримання яких не суперечить законодавству);

– інформування населення та проведення навчання (постійні професійні підготовки і підвищення кваліфікації працівників) з питань охорони праці;

– забезпечення координації діяльності органів державної влади, установ, організацій, об'єднань громадян, що розв'язують проблеми охорони здоров'я, гігієни та безпеки праці,

– організація співробітництва між роботодавцями та працівниками (їх представниками); між усіма соціальними групами під час прийняття рішень з охорони праці на місцевому та державному рівнях;

– світовий досвід у організації роботи щодо поліпшення умов і підвищення безпеки праці на основі міжнародного співробітництва [40,41] .

Робочим інструментом в ІТ-сфері, що є однією з найбільш перспективних галузей економіки, є комп’ютер з наявним на ньому програмним забезпеченням. Базовими основоположними законодавчими актами по охороні праці при роботі з ПК в Україні є НПАОП 0.00-7.15-18 "Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями", ДСанПіН 3.3.2.007-98 "Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин" та ДСТУ 8604:2015 "Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги", які призначені для запобігання впливу на працівників шкідливих і небезпечних факторів. До загально правової бази, що регулюють нормативно-

правові положення з техніки безпеки (й для ІТ-сфері в тому числі), відносяться Інструкції з охорони праці при роботі з електронно-обчислювальними машинами, в яких протокольно описано поводження під час штатних робіт та в аварійних ситуаціях, основним призначенням яких є запобігання наслідків (зменшення або повне їх усунення) небезпек, що пов’язані з виробничою діяльністю.

8.2 Шкідливі і небезпечні фактори при роботі з комп’ютером

Якщо послуговуватися класифікацією небезпечних й шкідливих виробничих факторів при роботі на комп’ютері по природі їх виникнення, а отже поділивши їх на наступні групи: фізичні, хімічні, психофізіологічні, біологічні, то до фізичних можемо віднести:

– підвищену й знижену температуру повітря;

– підвищену й знижену вологість повітря;

– недостатню освітленість робочого місця;

– перевищуючі припустимі норми шуму;

– підвищений рівень іонізуючого випромінювання;

– підвищений рівень електромагнітних полів;

– підвищений рівень статичної електрики;

– небезпеку враження електричним струмом;

– блякліть екрана дисплея;

до хімічних можемо віднести:

– виникнення, у результаті іонізації повітря при роботі комп'ютера, активних часток;

до психофізіологічних можемо віднести:

– розумова напруга;

– втрата реальності;

– виникнення залежності;

– нервово-емоційні перевантаження;

– перенапруга зорового аналізатора;

до біологічних можемо віднести:

– бактеріологічну небезпеку, пов’язану з наявністю місць із сприятливим середовищем їх розмноженням (наприклад клавіатура).

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Проаналізуємо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	4
Довжина	5
Висота	3

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м²	Не менше 6.0	6,67
Об'єм, V	м³	Не менше 20.0	20

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

У зазначеному приміщенні працюють троє людей. За даними, які наведено у табл. 8.1, та табл. 8.2, можна зробити висновок, що площа та об’єм приміщення з розрахунку на одне робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 "Державні санітарні правила і норми роботи з

візуальними дисплейними терміналами електронно-обчислювальних машин" [42] але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 "Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями" [42] та НПАОП 0.00-7.15-18 "Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями". Таким чином, можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Воло-гість, %	Швидкість повітря, м/с	Темпера-тура, °С	Воло-гість, %	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	40-55	0,11
Тепла	23-25	50-70	0,1	24-25	50-65	0,17

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер HP LaserJet Pro P1102, електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини та попереджають травматизм і професійні захворювання програмістів, є освітлення на робочому місці.

З 2019 року діють Державні будівельні норми України ДБН В.2.5-28:2018 "Природне і штучне освітлення" [43], у яких прописані вимоги до використання всіх освітлювальних приладів, у т.ч. світлодіодних.

Працю працівника, який постійно працює за комп’ютером, згідно ДБН В.2.5-28:2018 [43], можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення

коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [43], Крім того, все поле зору повинне бути освітлено достатньо рівномірно – це основна гігієнічна вимога. Так, як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

– розміри приміщення, у розрахунку на одному працюючого, відповідають нормативам;

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значень;

Таким чином, можна припустити, що основною причиною зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці й відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов’язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

Регулярне наочне знайомство персоналу із шляхами для евакуації людей із приміщення, відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками із заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так, як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

8.5 Розрахункова частина

Завдання: Розрахувати час для евакуації людей із виробничих приміщень.

Вхідні дані:

Виробниче приміщення, яке за вибухопожежною та пожежною небезпекою належить до категорії В, розташоване в одноповерховій будівлі зі ступенем вогнетривкості І. Один центральний поздовжній проїзд шириною 3 м та два поперечних проходів шириною 1 м поділяють приміщення на дві ділянки. З одного боку проїзду встановлено ворота з дверима для проходу людей, які в умовах вимушеної евакуації відіграють роль евакуаційного виходу. У найчисленнішій зміні працює 150 працівників.

Визначити відповідність заходів евакуації людей із виробничого приміщення встановленим нормам пожежної безпеки та розрахувати можливий час евакуації.

Початкові дані:

– кількість працівників у найчисленнішій зміни – 150;

– довжина цеху a = 50 м;

– ширина цеху b = 20 м;

– висота цеху h = 3 м.

Кількість працівників, які можуть опинитися в цеху у службових справах, становить 20 % від загальної кількості працівників. Виконаємо розрахунок.

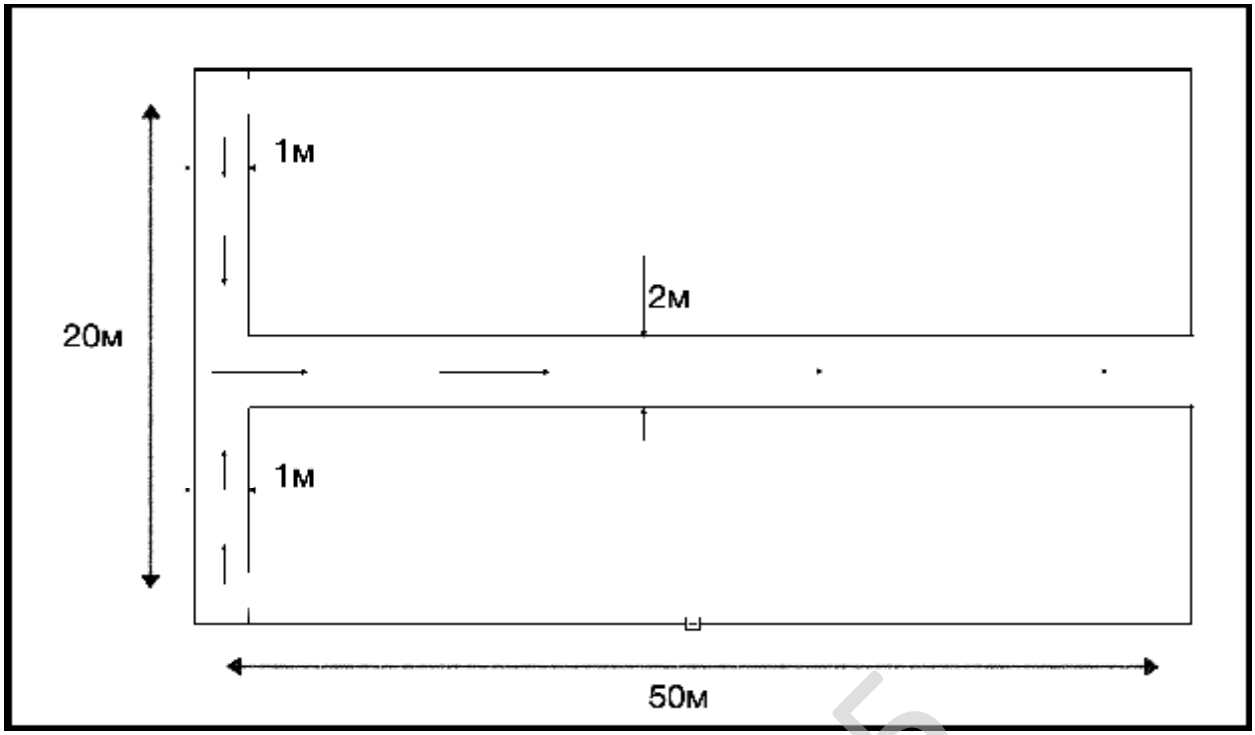

Рисунок 8.1 – Евакуаційний план

1. Визначаємо загальну кількість працівників, які можуть перебувати в цеху:

$$150 + 0{,}2 \cdot 150 = 180.$$

2. Оскільки прохід один, то на вихід припадає 180 працівників.

3. Найвіддаленішими від евакуаційних виходів є робочі місця, відстані від яких до евакуаційних виходів однакові й становлять 60 м (10 + 50).

4. Перевіримо, чи відповідає це значення нормативним даним, регламентованим СНиП 2.09.02-85. Для цього визначимо щільність людського потоку в загальних проходах Z. Оскільки на проходи припадає 180 працівників, а їх площа від робочих місць S = 10 ·1· 2 + 50 · 2 = 120 м^2, щільність людського потоку становитиме

$$Z = \frac{N}{S} = \frac{180}{120} \approx 2{,}0 \text{ працівника/м}^2$$

Відповідно до СНиП 2.09.02-85 максимально допустима відстань від найвіддаленішого робочого місця до евакуаційного виходу з приміщення

категорії В та вогнетривкості І, при такому значенні Z та об'ємі цеху 1000 м^3 становить 60 м. У даному випадку ця вимога виконується.

5. Визначимо необхідну ширину евакуаційного виходу, якщо відомо, що нормована кількість працівників на 1 м ширини такого виходу приміщення категорії В та вогнетривкості І становить 110, а на вихід припадає 180 працівників: В = 180/110 = 1,63 м; це відповідає вимозі, тому що ширина воріт, яка дорівнює ширині проїзду, становить 2 м.

6. Визначимо розрахунковий час евакуації з виробничого приміщення $t_{евак.розр}$, урахувавши, що він буде однаковий для обох ділянок. Таким чином,

$$t_{евак.\ розр} = t1 + t2$$

де t1, t2 – час евакуації працівників ділянок відповідно проходом і проїздом до виходу, хв.

7. Визначимо щільність людського потоку у проході та проїзді виробничого приміщення за формулою:

$$D1 = \frac{N1 \cdot f}{l1 \cdot \delta 1} = \frac{(180/2) \cdot 0{,}125}{10 \cdot 1} = 1{,}13 \text{ м}^2/\text{м}^2$$

$$D2 = \frac{N2 \cdot f}{l2 \cdot \delta 2} = \frac{180 \cdot 0{,}125}{50 \cdot 2} = 0{,}23 \text{ м}^2/\text{м}^2$$

За одержаними значеннями визначаємо швидкість людського потоку у проході (v1 = 15 м/хв) та проїзді цеху (v2 = 60 м/хв).

$$t1 = \frac{l1}{v1} = \frac{10}{15} = 0{,}66$$

$$t2 = \frac{l2}{v2} = \frac{50}{60} = 0{,}83$$

$$t_{евак.\ розр.} = t1 + t2 = 0{,}66 + 0{,}83 = 1{,}49 \text{ хв.}$$

Таким чином, розрахунковий час евакуації працівників із виробничого приміщення становить близько 1,5 хв.

Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз умов праці, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок часу евакуації зі службового приміщення. Розроблено заходи з охорони праці.

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

– Досліджена система захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

– На основі отриманих результатів досліджень створена програмна реалізація системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android.

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

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

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

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

Програма призначена для виконання під управлінням багатозадачної операційної системи Android.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм CRYPTON.

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.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бобровський Д.О. Дослідження та програмна реалізація системи захисту даних GPRS-мережі, розгорнутої на пристроях, які працюють під ОС Android // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.

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

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

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

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

6. Alex Matrosov, Eugene Rodionov, Sergey Bratus. Rootkits and Bootkits. No Starch Press. 2019. 450 p.

7. 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.

8. 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

9. 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.

10. 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.

11. 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.

12. 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.

13. 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.

14. 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.

15. 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.

16. 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.

17. 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.

18. 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.

19. 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.

20. 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.

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

22. 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

23. 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.

24. 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.

25. 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.

26. 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.

27. Смірнов О.А. Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *ІІ 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.

28. 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.

29. 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.

30. 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.

31. 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,

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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

38. 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.

39. 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.

40. 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.*

41. 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.

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

43. 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.

44. 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.

45. 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.

46. 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.

47. 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.

48. 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.

49. 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.

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

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