

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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи забезпечення безпеки
корпоративної мережі від зловмисних дій”

Виконав здобувач вищої освіти
IV курсу, групи КІ-22-2
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Невгоденко В.І.
«___» _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Марченко К.М.
«___» _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань 12 "Інформаційні технології"
Спеціальність 123 "Комп'ютерна інженерія"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

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

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Невгоденку Вадиму Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи забезпечення безпеки корпоративної мережі від зловмисних дій*

2. Керівник роботи *Марченко Костянтин Миколайович, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 136-02 від 27.02.2026 року

3. Строк подання студентом роботи до захисту 23.05.2026 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи забезпечення безпеки корпоративної мережі від зловмисних дій*

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

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

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

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

7. Дата видачі завдання « 27 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника

Марченко К.М.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Невгоденко В.І.
(прізвище та ініціали)

АНОТАЦІЯ

Невгоденко В.І. Програмне забезпечення системи забезпечення безпеки корпоративної мережі від зловмисних дій. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи забезпечення безпеки корпоративної мережі від зловмисних дій.

Метою розробки є програмне забезпечення системи забезпечення безпеки корпоративної мережі від зловмисних дій.

Результат роботи – програмна реалізація системи забезпечення безпеки корпоративної мережі від зловмисних дій.

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 ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Visual C#.

Ключові слова: комп'ютерна інженерія, безпека корпоративної мережі

ABSTRACT

Nevhodenko V.I. Software for the corporate network security system against malicious actions. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed, which is intended for the corporate network security system against malicious actions.

The purpose of the development is software for the corporate network security system against malicious actions.

The result of the work is the software implementation of the corporate network security system against malicious actions.

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 PC with Windows 10/11.

The program was developed in the Visual C# environment.

Keywords: computer engineering, corporate network security

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	4
1.1 Призначення системи.....	4
1.2 Область застосування.....	5
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	6
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	6
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	17
2.3 Розгорнута постановка завдання	20
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	21
3.1 Опис функціонування системи	21
3.2 Розробка структурної схеми.....	28
3.3 Розробка функціональної схеми	45
3.4 Розробка діаграми процесів.....	57
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	59
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	59
4.2 Захист розробленого програмного забезпечення.....	66
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	70
6 ОСНОВНІ ВИСНОВКИ.....	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	74

					ВКРБ-123.26.0053.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	<i>Програмне забезпечення системи забезпечення безпеки корпоративної мережі від зловмисних дій</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>		<i>Невгоденко В.І.</i>				Б		
<i>Перев.</i>		<i>Марченко К.М.</i>					1	81
<i>Н.контр.</i>		<i>Коваленко А.С.</i>				ЦНТУ КІ-22-2		
<i>Затв.</i>		<i>Смірнов О.А.</i>						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АРМ	—	автоматизоване робоче місце
ЕЦП	—	електронний цифровий підпис
ЗКЦ	—	засвідчуючий й ключовий центр
ЗЦ	—	засвідчуючий центр
КЦ	—	ключовий центр
ЛОМ	—	локальна обчислювальна мережа
СКЗІ	—	система контролю та захисту інформації
ЦР	—	центр реєстрації
ЦУМ	—	центр управління мережею
SOA	—	Service-Oriented Architecture
SOAP	—	Simple Object Access Protocol

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Кіберзлочинці все частіше атакують пристрої у корпоративної мережі, що забезпечують роботу бізнесу. Зараз захист цих кінцевих точок – ноутбуків, настільних комп'ютерів, смартфонів і серверів, а також мережевого обладнання – не є варіантом, а головним пріоритетом. Чи знаєте ви, що лише у 2025 році колосальні 68 відсотків організацій зазнали принаймні однієї атаки на кінцеві точки, що призвела до компрометації даних або ІТ-інфраструктури? Така цифра показує гостру потребу в передових продуктах для захисту кінцевих точок корпоративної мережі, щоб виявляти, запобігати та знищувати кіберзагрози на кожній точці входу.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи забезпечення безпеки корпоративної мережі від зловмисних дій.

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

- Огляд існуючих систем забезпечення безпеки корпоративної мережі від зловмисних дій.
- Дослідження системи забезпечення безпеки корпоративної мережі від зловмисних дій.
- Програмна реалізація системи забезпечення безпеки корпоративної мережі від зловмисних дій.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі забезпечення безпеки корпоративної мережі від зловмисних дій.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи забезпечення безпеки корпоративної мережі від зловмисних дій, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для забезпечення безпеки корпоративної мережі від зловмисних дій. Це необхідно для забезпечення безпеки інформації, яка передається по системі, побудованій з використанням концепції сервіс-орієнтованої архітектури.

План реалізації безпеки SCA-додатку складається з виконання наступних кроків:

- Сформуванати оптимальну робочу групу.
- Створити деталізований план проекту.
- Вести таблицю рішень по безпеці для процесу впровадження SCA.
- Створити проект використання принципу "безпека зсередини" і інфраструктури керування ризиками.
- Визначити зацікавлених осіб усередині організації ("своїх") і за її межами (сторонніх).
- Правильно вибрати й використовувати інструменти для збору вимог.
- При реалізації безпеки SCA-додатку дотримуватися схеми процесу SDLC.
- Ознайомитися з існуючими моделями й навчитися на їхньому прикладі.
- Застосувати стандарти WS-Security.
- Розробити стандарти для сторонніх розроблювачів.

Виконавши всі ці дії, ви забезпечите собі гарну стартову точку в реалізації безпеки SCA-додатку.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1.2 Область застосування

Областю застосування системи є компонентна архітектура сервісів (SCA). Архітектура компонентів служб (SCA) складається з ряду специфікацій, що описують модель компонування додатків і систем у концепції архітектури на основі служб (SOA).

Архітектура на основі служб (SOA) – це інфраструктура, що поєднує окремі бізнес-функції й процеси, називані службами, у комплексні бізнес-додатки. В інфраструктурі SOA узагальнені бізнес-компоненти представлені службами. SOA дозволяє структурувати ресурси ІТ у вигляді повторно використовуваних служб, які нежорстко зв'язані між собою й не залежать від платформи й реалізації. В SOA рішення представляються у вигляді комплексу служб, які зв'язані між собою за допомогою чітко певних інтерфейсів і контрактів.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи забезпечення безпеки корпоративної мережі від зловмисних дій, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

У даному розділі розглядаються найкращі продукти для захисту кінцевих точок корпоративної мережі у 2026 році, з вичерпним порівнянням, що детально описує функції, моделі ціноутворення та оцінки користувачів. Ми обговоримо найкращі рекомендації, пояснимо, як вибрати найбільш підходящий продукт для вашої організації, та відповімо на поширені запитання користувачів, щоб зміцнити ваш захист від постійно мінливого ландшафту кіберзагроз.

Продукти для захисту кінцевих точок корпоративної мережі – це високорозвинені програмні рішення, які захищають окремі пристрої від кіберзагроз, таких як шкідливе програмне забезпечення, програми-вимагачі та АРТ. Ці рішення зазвичай поєднують кілька технологій безпеки, таких як антивірус, брандмауер, шифрування даних та аналіз поведінки, для комплексного захисту.

Згідно зі звітом IBM про вартість витоку даних за 2025 рік, середні витрати на витік даних зараз становлять 4,88 мільйона доларів. Ця статистика багато говорить про фінансові наслідки неналежного захисту кінцевих точок. Як наслідок, безпека кінцевих точок стала дедалі більшою проблемою в сучасному бізнес-середовищі, з широкими можливостями для атак, що виникають внаслідок віддаленої роботи та хмарних сервісів.

Динаміка цифрового світу змінюється щодня, як і тактика кіберзлочинців. Саме тому надійна безпека кінцевих точок – це сфера забезпечення безпеки корпоративної мережі від зловмисних дій, яку не можна скомпрометувати в жодній стратегії.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Кілька критичних факторів підкреслюють необхідність надійної безпеки кінцевих точок корпоративної мережі:

– Поширення кінцевих точок: Все більше співробітників працюють віддалено, і більшість із них впроваджують культуру BYOD. Це означає, що кожна організація зараз має кінцеві точки, ніж будь-коли раніше. Кожен пристрій є потенційною точкою входу для кібератаки, і всі вони мають бути захищені.

– Витонченість кіберзагроз: Кіберзлочинці не припиняють розробляти нові тактики та витончені методи для зламу корпоративних мереж. Продукти безпеки кінцевих точок корпоративної мережі, у свою чергу, повинні йти в ногу з такими загрозами, пропонуючи проактивний захист від відомих та невідомих вразливостей у середовищі організації.

– Захист даних та дотримання вимог: Суворі правила захисту персональних даних включають h. Безпека кінцевих точок передбачає значні штрафи за порушення даних, що дуже важливо для отримання такої конфіденційної інформації та, як наслідок, для допомоги організаціям у дотриманні цих правил.

– Перехід до віддаленої роботи: Перехід до віддаленої роботи, разом з іншими змінами, призвів до руйнування мережевого периметра, такого дорогого для менеджерів з безпеки, що значно ускладнило захист корпоративних активів. Продукти безпеки кінцевих точок корпоративної мережі пропонують життєво важливий рівень захисту для пристроїв, що використовуються поза межами традиційного офісного сегмента.

– Вартість кіберзлочинності: Кіберзлочинність має значні фінансові витрати. За даними IBM, середня ціна витоку даних досягла 4,88 мільйона доларів у 2024 році. Надійна безпека кінцевих точок може значно знизити цей ризик і захистити прибуток організації.

Кожне рішення пропонує різні функції та можливості, досліджуючи найкращі продукти для безпеки кінцевих точок корпоративної мережі у 2026 році. Ми розглянемо їх та надамо огляд. Для отримання додаткової інформації ви

можете переглянути їхні рейтинги та огляди Gartner. Почнемо.

SentinelOne

SentinelOne пропонує захист кінцевих точок наступного покоління, використовуючи штучний інтелект та машинне навчання для захисту ноутбуків, настільних комп'ютерів та мобільних пристроїв. Він може захистити всі аспекти ваших кінцевих точок та керувати поверхнями атаки. Ви можете отримати автоматичний захист, виявлення загроз та можливості реагування на різні кіберзагрози. У міру масштабування вашої організації функції захисту кінцевих точок SentinelOne будуть відповідати змінам середовища загроз; забронюйте безкоштовну демонстрацію, щоб дізнатися більше.

Огляд платформи:

1. Singularity™ Cloud Security – Singularity™ Cloud Security використовує машинне навчання для виявлення та видалення загроз кінцевим точкам у режимі реального часу, будь то програми-вимагачі чи атаки нульового дня. Він негайно розпізнає шкідливе програмне забезпечення, щоб мінімізувати ризик порушення, перш ніж може бути завдано значної шкоди. Він також автоматизує реагування на загрози кінцевим точкам шляхом стримування та усунення атак протягом кількох секунд, без участі людини. Безагентна CNAPP від SentinelOne надає єдину консоль для керування всією безпекою кінцевих точок, завдяки якій адміністратори можуть бачити стан пристроїв, застосовувати політики безпеки та переглядати звіти з однієї інформаційної панелі. Це спрощує керування кінцевими точками як у хмарних, так і в локальних середовищах.

2. Singularity™ XDR – Singularity™ XDR розширює захист кінцевих точок, що взаємодіють із хмарою. Він забезпечує захист кінцевих точок від програм-вимагачів, безфайлових атак та атак «нульового дня»; він захищає всі ваші кінцеві точки в хмарі та локально від загроз наступного покоління. Він максимізує видимість у всіх сферах підприємства та забезпечує поєднання неперевершеного покриття та швидкості виявлення.

3. Singularity™ Threat Intelligence – Singularity™ Threat Intelligence

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

забезпечує проактивне розуміння ваших кінцевих точок, генеруючи аналітичні дані про загрози світового класу. Ви можете активно відстежувати та зменшувати ризики, а також отримувати практичну інформацію для виявлення зловмисників на кінцевих точках. Аналітика зловмисників SentinelOne працює на базі Mandiant, а платформа також забезпечує 200 000 годин реагування на інциденти на рік. Ви також можете сортувати сповіщення безпеки з урахуванням контексту зловмисника.

Особливості:

– Захищає кінцеві точки, сервери та мобільні пристрої – SentinelOne надає можливості запобігання, виявлення та реагування на атаки в масштабах усієї організації. Він інформує вас про різні поверхні атак, централізує дані та робочі процеси, а також відповідно керує всіма парками пристроїв.

– Динамічне виявлення пристроїв: Автоматично інвентаризуйте свої кінцеві точки та ресурси хмарної безпеки. Завдяки автономному комбінованому рішенню EPP+EDR ви можете зменшити кількість хибнопозитивних результатів та послідовно підвищити ефективність виявлення в різних ОС.

– Швидка реакція та час окупності: одним клацанням миші виправляйте та відкочуйте кінцеві точки, скорочуючи час реагування та пришвидшуючи розслідування.

– Контроль поверхні мережевих атак – знаходьте та відмічайте відбитки всіх IP-пристроїв у вашій мережі за допомогою Ranger. Зрозумійте ризики, які вони становлять, та автоматично розширюйте захист.

– Відповідність багатохмарним стандартам: Забезпечення відповідності найновішим нормативним стандартам та системам безпеки, включаючи SOC 2, HIPAA, PCI-DSS, NIST та CIS Benchmark.

Основні проблеми, які усуває SentinelOne:

– Загрози нульового дня: SentinelOne, завдяки своєму підходу на основі штучного інтелекту, також може виявляти та нейтралізувати раніше невідомі загрози для організацій, які можуть мати слабкі місця в уразливостях нульового

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

дня.

- Ручне стримування загроз: SentinelOne автоматизує виявлення та усунення загроз, звільняючи команду безпеки для роботи над стратегічними ініціативами, а не витрачаючи час, необхідний для ручного втручання.

- Уніфікація управління безпекою: Уніфікована консоль значно спрощує операції безпеки. Вона зменшує складність зберігання різних інструментів безпеки та підвищує ефективність і зручність використання: ця функція забезпечує глибокий огляд діяльності кінцевих точок, дозволяючи організації розуміти та виправляти всі потенційні прогалини в безпеці в мережі.

- Усуває складні постійні загрози – може боротися з атаками соціальної інженерії, інсайдерами, а також прихованими та невідомими загрозами.

CrowdStrike Falcon

Falcon від CrowdStrike – це хмарний продукт. Це антивірус наступного покоління, який поєднує можливості роботи з кінцевими точками та реагування на них з розвідкою загроз в одному пакеті для організацій будь-якого розміру.

Особливості:

- Безперебійне розгортання: хмарне розгортання та хмарна архітектура рішень усувають потребу в локальній інфраструктурі.

- Аналітика загроз у режимі реального часу на основі даних, зібраних з краудсорсингу: система збирає дані з мільйонів кінцевих точок по всьому світу, забезпечуючи актуальну аналітику загроз та проактивний захист у режимі реального часу.

- Поведінкова аналітика може виявляти складні атаки. Використовуючи машинне навчання та штучний інтелект, вона аналізує поведінку кінцевих точок, щоб виявити аномалії, пов'язані з потенційними складними загрозами.

- Проактивне виявлення: Полювання на загрози надає командам безпеки інструменти в певний момент часу, за допомогою яких можна виявити їхній проактивний пошук невідомих загроз та вразливостей, пов'язаних з мережею.

- Інтегрована аналітика загроз та управління вразливостями: цей процес

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

агрегує та сканує дані для визначення пріоритетів та усунення значних прогалин у безпеці.

Безпека кінцевих точок Symantec

Broadcom володіє Symantec Endpoint Security, яка пропонує пакети можливостей захисту кінцевих точок, включаючи машинне навчання, поведінковий аналіз та розвідку загроз.

Особливості:

- Розширене машинне навчання: воно використовує складні алгоритми, які покращують можливості ідентифікації та блокування відомих та невідомих шкідливих програм за допомогою розширеного виявлення загроз.
- Інтегроване виявлення та реагування на кінцеві точки: це дозволяє виявляти, досліджувати та реагувати на загрози кінцевих точок.
- Можливості контролю пристроїв та програм: це дозволяє встановлювати політики, що регулюють доступ до пристроїв, і одночасно дозволяє регулювати програми для обмеження ризиків безпеки.
- Мережевий брандмауер та засоби запобігання вторгненням запобігають вторгненням на мережевому рівні, моніторячи та блокуючи підозрілий мережевий трафік.
- Інтеграція захисту від мобільних загроз розширює захист на мобільні пристрої для протидії загрозам та вразливостям, характерним для мобільних пристроїв.

Дізнайтеся більше про можливості захисту кінцевих точок Symantec, переглянувши рейтинги та огляди Gartner і TrustRadius.

Microsoft Defender for Endpoint

Microsoft Defender for Endpoint – це рішення безпеки від Microsoft, призначене для підприємств. Воно виявляє, досліджує та усуває загрози в мережі та на кількох кінцевих точках в організації.

Особливості:

- Хмарний захист: використовує глобальну інфраструктуру Microsoft для

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

забезпечення захисту в режимі реального часу від нових загроз – на основі аналітики загроз Microsoft.

- Можливості управління загрозами та вразливостями: виявляє, визначає пріоритети та класифікує відомі загрози та неправильні конфігурації на кінцевих точках на основі їхніх вразливостей.

- Автоматизоване виявлення та реагування: Автоматизує процеси виявлення та реагування, пов'язані із загрозами, для зменшення трудомісткості діяльності та підвищення ефективності.

- Розширене полювання для проактивного виявлення загроз: це дозволяє командам безпеки проактивно виявляти загрози, використовуючи підхід на основі запитів.

- Він добре інтегрується з іншими продуктами безпеки Microsoft, включаючи Microsoft 365 та служби Azure, для створення інтегрованої системи безпеки.

Trend Micro Vision One

Цей випуск Trend Micro Vision One включає систему виявлення та реагування на загрози, що доповнює безпеку кінцевих точок, яку забезпечує лише один агент.

Особливості:

- XGen Security: XGen Security пропонує кілька підходів до виявлення, включаючи найсучасніші технології машинного навчання та поведінкову аналітику, для більш інноваційного виявлення.

- Віртуальне виправлення: Віртуальне виправлення усуває вразливості до випуску офіційних патчів, захищаючи більш дійсні кінцеві точки від відомих та невідомих вразливостей.

- Підключений захист від загроз із покращеною видимістю: інтегрується з іншими рішеннями Trend Micro для обміну інформацією про загрози та зміцнення загальної безпеки.

- Можливості EDR: Пропонує можливості виявлення та реагування на

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

кінцеві точки для виявлення та належного стримування загроз на рівні кінцевих точок.

– Хмарна консоль: Хмарна консоль спрощує централізоване керування та перегляд звітів.

Sophos Intercept X

Sophos Intercept X – це рішення для захисту кінцевих точок корпоративної мережі, яке забезпечує організаціям захист від експлойтів, програм-вимагачів, виявлення шкідливого програмного забезпечення глибокого навчання та безпеку.

Особливості:

– Виявлення шкідливого програмного забезпечення за допомогою глибокого навчання: воно використовує нейронні мережі, які виявляють та блокують шкідливе програмне забезпечення, не покладаючись на методи виявлення на основі сигнатур.

– Запобігання експлойтам: Блокує метод, що використовується в експлойті, шляхом ідентифікації до того, як експлойт зможе бути виконаний.

– Пом'якшення наслідків активного супротивника: Виявляє та запобігає ворожій поведінці, яка може бути діяльністю активного супротивника.

– Захист від програм-вимагачів: продукт оснащений функцією CryptoGuard, яка запобігає несанкціонованому шифруванню файлів.

– Аналіз першопричин: надає детальну інформацію про те, як сталася атака, допомагаючи запобігти майбутнім інцидентам.

VMware Carbon Black Cloud

VMware Carbon Black Cloud поєднує антивірус, виявлення кінцевих точок, реагування та пошук загроз в одному місці за допомогою хмарної платформи захисту кінцевих точок.

Особливості:

– Запобігання потоковій передачі для захисту від загроз у режимі реального часу: Активність кінцевих точок відстежується проактивно для швидкого виявлення та запобігання загрозам у режимі реального часу.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

- Поведінкове EDR для розширеного виявлення загроз: Поведінкове EDR виявляє зловмисну поведінку, яка може обійти інші традиційні форми захисту.
- Корпоративне EDR для пошуку загроз та реагування на інциденти: Розширені інструменти безпеки для розслідування та реагування на інциденти для команд безпеки.
- Контроль програм для розширеного захисту системи: Це контролює роботу програм, запобігаючи запуску небажаного програмного забезпечення в системі.
- Хмарна архітектура: простіше та безперешкодніше розгортати та масштабувати в різних середовищах.

Вибір відповідного продукту для захисту кінцевих точок корпоративної мережі вашої організації може бути важливим, але складним рішенням, яке має далекосяжний вплив на загальний стан вашої забезпечення безпеки корпоративної мережі від зловмисних дій. З огляду на це, ось повний посібник для прийняття мудрого рішення:

- Оцінка потреб вашої організації: Першим кроком у виборі ідеального рішення є оцінка ваших потреб у сфері безпеки. Слід враховувати розмір вашого бізнесу, типи кінцевих точок, галузеві вимоги до відповідності та поточну інфраструктуру безпеки. Це допоможе вам оцінити найважливіші функції та можливості.
- Проаналізуйте основні характеристики: Шукайте продукти для захисту кінцевих точок корпоративної мережі із широким набором функцій, які відповідають вашим потребам. Під час оцінювання враховуйте такі ключові характеристики:
 - Розширене виявлення та запобігання загрозам: Штучний інтелект з машинним навчанням для виявлення загроз.
 - EDR, або виявлення та реагування на кінцеві точки, – це важлива функція, яку повинні мати всі продукти безпеки кінцевих точок корпоративної мережі. Вона також використовується для пошуку та розслідування загроз.

- Інтеграція розвідки про загрози в режимі реального часу продовжує допомагати їм бути в курсі нових загроз.
- Автоматизоване реагування та усунення наслідків: це автоматизує нейтралізацію загроз без залучення людини.
- Централізоване управління та звітність: це забезпечить прозорість у всіх сферах завдяки нагляду та звітності про дотримання вимог.

Вибираючи рішення, ви повинні переконатися, що воно забезпечує вашому бізнесу збалансоване поєднання проактивного захисту, виявлення та реагування. Ось що ще потрібно врахувати:

- Зважте на масштабованість та гнучкість: Розгляньте рішення, яке може зростати разом з вашою організацією. Більшість хмарних продуктів для безпеки кінцевих точок корпоративної мережі є більш масштабованими та гнучкими, ніж інші. Ви можете легко додавати або видаляти кінцеві точки, коли ваш бізнес змінюється. Виберіть рішення, яке підходить для різних операційних систем та типів пристроїв.

- Аналіз продуктивності: Рішення для захисту кінцевих точок корпоративної мережі не повинно суттєво впливати на продуктивність, пропонуючи при цьому всі переваги надійного захисту. Шукайте легкі агенти та оптимізацію ресурсів. Продукт може сприяти його оцінці з точки зору очікуваного впливу на продуктивність.

- Простота розгортання та управління: Простота розгортання та постійного управління призведуть до значної економії загальної вартості володіння. Продукти, які спрощують процес розгортання, мають зручні консолі управління та забезпечують шляхи для автоматичних оновлень, пройшли цей тест. Централізовані можливості управління можуть оптимізувати адміністрування великих або розподілених середовищ.

- Огляд можливостей аналізу загроз: Ефективна безпека кінцевих точок залежить від поточної інформації про загрози. Проаналізуйте якість та широту мережі аналізу загроз кожного постачальника. Рішення, що базуються на

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

машинному навчанні та штучному інтелекті для аналізу глобальних даних, можуть запропонувати набагато кращий захист.

– Дослідіть можливості звітності та аналітики. Розуміння стану та відповідності вимогам є важливим, тому знайдіть рішення, які пропонують розширену звітність та аналітику за допомогою налаштовуваних інформаційних панелей, журналів подій та звітів, що стосуються відповідності.

– Розгляньте можливості інтеграції: Забезпечте бездоганну інтеграцію всіх інших елементів інфраструктури безпеки з безпекою кінцевих точок – дізнайтеся, як кожен продукт інтегрується з SIEM, рішеннями для керування доступом до ідентифікаційних даних або іншими інструментами безпеки.

– Підтримка та ресурси постачальників: Порівняйте підтримку та ресурси, які пропонує кожен постачальник. Постачальник рішень повинен надавати детальну документацію, швидку підтримку клієнтів та доступ до досліджень загроз і передового досвіду. Постачальники з широкими програмами навчання та сертифікації краще допоможуть вашим командам отримати максимальну рентабельність інвестицій від рішення.

– Аналіз загальної вартості володіння: Хоча ціна на момент покупки є актуальною, визначте, скільки коштуватиме володіння протягом певного часу. Окрім ліцензійних зборів, враховуйте необхідні витрати на обладнання, навчання персоналу та інші поточні витрати на обслуговування. Рішення можуть бути дорогими на початку, але забезпечують довгострокову економію завдяки зниженню накладних витрат на управління або кращому захисту.

Оскільки загрози стають дедалі складнішими та частішими, рішення для захисту кінцевих точок корпоративної мережі повинні бути центральним елементом комплексної стратегії забезпечення безпеки корпоративної мережі від зловмисних дій організації. Розглянуті вище рішення є одними з найкращих продуктів для захисту кінцевих точок корпоративної мережі, доступних у 2026 році. Вибираючи правильний продукт для захисту кінцевих точок корпоративної мережі, бізнес повинен враховувати різні особливості, такі як масштабованість,

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

вплив на продуктивність, простота керування та можливості інтеграції.

Суть полягає в тому, що ідеальний продукт для захисту кінцевих точок корпоративної мережі повинен відповідати вашій організації та інфраструктурі. Максимально використовуйте безкоштовні пробні версії та демонстрації від постачальників, щоб отримати практичне уявлення про те, як кожне рішення працюватиме у вашому середовищі. Зрештою, ви повинні зробити крок до покращення рівня безпеки вашої організації.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Програмне забезпечення написане мовою Visual C#. Ця мова обрана виходячи з наступних міркувань. Visual C# – строго типізована об'єктно-орієнтована мова, призначена для розробки різноманітних безпечних і потужних додатків, виконуваних у середовищі .NET Framework. Мовою Visual C# можна розробляти звичайні клієнтські додатки Windows, веб-служби XML, розподілені компоненти, додатки типу “сервер-клієнт”, додатки баз даних і багато яких інших. В Visual C# є розширений редактор коду, конструктори зі зручним користувацьким інтерфейсом, вбудований відладник і багато інших засобів, покликані спростити розробку додатків мовою Visual C# версії 5.0 і .NET Framework версії 4.5.

Синтаксис Visual C# дуже виразний, але простий у вивченні. Усі, хто знаком з мовами C, C++ або Java з легкістю визнають синтаксис із фігурними дужками, характерний для мови Visual C#. Розроблювачі, що знають кожну із цих мов, як правило, зможуть домогтися ефективної роботи з мовою Visual C# за дуже короткий час. Синтаксис Visual C# робить простіше те, що було складно в C++, і забезпечує потужні можливості, такі як типи значень Nullable, перерахування, делегати, лямбда-вираження й прямий доступ до пам'яті, чого немає в Java. Visual C# підтримує універсальні методи й типи, забезпечуючи

більше високий рівень безпеки й продуктивності, а також ітератори, що дозволяють при реалізації колекцій класів визначати власне поводження ітерації, що може легко використовуватися в клієнтському коді. В Visual C# 5.0 вираження LINQ (Language-Integrated Query) роблять строго-типізований запит першокласною конструкцією мови.

Як об'єктно-орієнтована мова, Visual C# підтримує поняття інкапсуляції, спадкування й поліморфізму. Всі змінні й методи, включаючи метод Main – точку входу додатка – інкапсуються у визначення класів. Клас може успадковувати безпосередньо з одного родового класу, але може реалізовувати будь-яке число інтерфейсів. Для методів, які перевизначають віртуальні методи в батьківському класі, необхідно ключове слово `override`, щоб виключити випадкове повторне визначення. У мові Visual C# структура схожа на полегшений клас: це тип, що розподіляється по стопках, що реалізує інтерфейси, але не підтримуюче спадкування.

На додаток до основних описаних об'єктно-орієнтованих принципів, мова Visual C# спрощує розробку компонентів програмного забезпечення завдяки декільком інноваційним конструкціям мови, у число яких входять наступні:

- Інкапсульовані підписи методів, називані делегатами, які підтримують строго-типізовані повідомлення про події.
- Властивості, що виступають у ролі методів доступу для закритих змінних-членів.
- Атрибути з декларативними метаданими про типи під час виконання.
- Вбудовані коментарі XML-документації.
- LINQ (Language-Integrated Query), що пропонує вбудовані можливості запитів у різних джерелах даних.

Якщо буде потрібно забезпечити взаємодію з іншим програмним забезпеченням Windows, таким як об'єкти COM або власні бібліотеки DLL Win32, у мові Visual C# можна використовувати процес, що називається "Interop". Процес Interop дозволяє програмам на Visual C# виконувати практично будь-які

дії, які може виконувати вихідний додаток на C++. Мова Visual C# підтримує навіть покажчики й поняття "небезпечного" коду для тих випадків, коли прямий доступ до пам'яті має вкрай важливе значення.

Процес побудови Visual C# у порівнянні з C і C++ простий і є більше гнучким, чим в Java. Немає окремих файлів заголовка, а методи й типи не потрібно повідомляти в певному порядку. У вихідному файлі Visual C# може бути визначене будь-яке число класів, структур, інтерфейсів і подій.

Архітектура платформи .NET Framework

Програма мовою Visual C# виконується в середовищі .NET Framework – інтегрованому компоненті Windows, що містить віртуальну систему виконання (середовище CLR) і уніфікований набір бібліотек класів. Середовище CLR являє собою комерційну реалізацію корпорацією Майкрософт інфраструктури CLI, що є міжнародним стандартом, який лежить в основі створення середовищ виконання й розробки, у яких забезпечується тісна взаємодія між мовами й бібліотеками.

Вихідний код, написаний мовою Visual C#, компілюється в проміжну мову (IL) у відповідності зі специфікацією CLI. Код IL і ресурси, такі як растрові зображення й рядки, зберігаються на диску у файлі, що виконується, названому складанням, з розширенням EXE або DLL у більшості випадків. Складання містить маніфест із відомостями про типи складання, версії, мови й регіональні параметри та вимоги безпеки.

При виконанні програми на Visual C# складання завантажується в середовище CLR залежно від відомостей у маніфесті. Далі, якщо вимоги безпеки дотримані, середовище CLR виконує JIT-компіляцію для перетворення коду IL в інструкції машинного коду. Середовище CLR також надає інші служби, що відносяться до автоматичного збору сміття, обробки виключень і керуванню ресурсами. Код, виконуваний середовищем CLR, іноді називають "керованим кодом" у протиставлення "некерованому коду", що компілюється в машинний код, призначений для певної системи.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи забезпечення безпеки корпоративної мережі від зловмисних дій.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) 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) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Система забезпечення безпеки корпоративної мережі від зловмисних дій рідко змінюється за одну ніч. Натомість вона непомітно розвивається, доки не буде досягнуто переломного моменту. До 2026 року цей переломний момент буде неможливо ігнорувати.

Історично більшість програм безпеки розроблялися з урахуванням стримування. Інцидент траплявся, інструмент його виявив, команда реагувала, і шкода обмежувалася визначеним середовищем. Така модель більше не відображає реальності.

Ця визначальна зміна формує тенденції забезпечення безпеки корпоративної мережі від зловмисних дій у 2026 році: кіберризик стає системним. Він більше не існує в межах одного контролю, однієї команди чи одного інструменту. Він виникає в результаті взаємодії між людьми, платформами, автоматизацією та швидкістю.

Людська помилка завжди була фактором кіберінцидентів. Найбільшою зміною в найближчі роки стане масштаб.

Штучний інтелект не винаходить абсолютно нових методів атак. Натомість він посилює існуючі: фішинг, видавання себе за іншу особу, соціальну інженерію, роблячи їх швидшими, дешевшими та складнішими для відмінності від законної поведінки. Атаки, які колись вимагали ручних зусиль, тепер можуть виконуватися безперервно, адаптуючись у режимі реального часу.

Галузеві дані підтверджують цю тенденцію до зростання. Звіт Palo Alto Networks Unit 42 про глобальне реагування на інциденти показує, що атаки на основі ідентифікації зараз є найпоширенішим вектором початкового доступу,

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

причому зловмисники все частіше покладаються на дійсні облікові дані, а не на шкідливе програмне забезпечення.

Водночас організації спостерігають швидке впровадження генеративних інструментів штучного інтелекту в повсякденні робочі процеси, часто без централізованого контролю. Це створює новий клас ризиків: конфіденційні дані, що передаються несанкціонованим інструментам, контент, згенерований штучним інтелектом, який використовується в бізнес-процесах, та автоматизовані дії, що виконуються без урахування контексту безпеки.

До 2026 року людський фактор стане проблемою автоматизації, а не обізнаності. Керівники служб безпеки повинні розробляти стратегії виявлення та реагування з урахуванням того, що обман буде успішним.

Розуміння нової парадигми є необхідністю для кожної стратегії безпеки підприємства, яка рухається вперед.

На відміну від минулого, сучасні кібератаки не рухаються лінійно. Натомість вони за лічені хвилини проникають через ідентифікаційні дані, хмарні сервіси, кінцеві точки, API та сторонні платформи. Часто для цього вони використовують легітимні облікові дані та довірені з'єднання. Результатом є не єдине, локалізоване порушення, а каскадне поширення впливу на кілька рівнів підприємства.

Лідери у сфері безпеки стикаються не з раптовою хвилею нових загроз, а з накопиченням структурних зрушень, які зростали роками: атаки, зумовлені автоматизацією, розростання ідентифікаційних даних, швидкість хмарних технологій та глибоко взаємопов'язані цифрові екосистеми. Це поєднання переосмислює те, як виникають, масштабуються та впливають на бізнес кіберризики.

У 2026 році ситуація з загрозами безпеці корпоративної мережі зростає, і зловмисники все частіше використовують легітимні інструменти як зброю, передові тактики соціальної інженерії та максимізують ефективність за рахунок повторного використання коду. Організації повинні бути готові до складних атак,

які використовують вроджену довіру та розмивають межі між законною та зловмисною діяльністю.

Зловживання легальним програмним забезпеченням

Прогнозується, що у 2026 році спостерігатиметься подальше зростання зловживання легітимним програмним забезпеченням, зокрема інструментами віддаленого керування та моніторингу (RMM). Зловмисники успішно перетворюють ці необхідні адміністративні утиліти на потужні канали для своїх кампаній, що дозволяє їм отримувати практичний доступ до цільових середовищ за допомогою клавіатури.

Цей метод є дуже ефективним, оскільки інструменти RMM часто є надійними, внесеними до білого списку та важливими для бізнес-операцій. Після встановлення контролю зловмисники проводять розвідку та збирають критично важливу інформацію про користувачів та машини, що створює умови для руйнівних атак на пізніших етапах, зокрема кампаній з вимагання (вимагання).

Деякі ілюстративні тематичні дослідження, які ми бачили цього року, включають:

– Експлуатація вразливості RMM: У лютому 2025 року було повідомлено про вразливість у SimpleHelp RMM. Хоча початкове виявлення захистило деяких клієнтів, наступна кампанія приблизно у травні 2025 року призвела до того, що картель викупу DragonForce успішно використав цю саму вразливість для атаки на постачальника керованих послуг (MSP) у Великій Британії, що підкреслює швидкість, з якою експлойти переходять від виявлення до використання як зброї.

– Підрив ланцюга поставок: Ще раз демонструючи проблему довіри до програмного забезпечення, дослідники CyberProof виявили бінарні файли ConnectWise ScreenConnect з дійсними цифровими підписами, які здійснюють вихідні з'єднання з підозрілими серверами командування та управління (C2). Ця атака включала Authenticode Stuffing – впровадження шкідливого коду зі збереженням цілісності оригінального підпису. Хоча сертифікат пізніше було відкликано, ці бекдор-дроппери згодом використовувалися іншими

зловмисниками для розповсюдження інформаційних програм, що підкреслює ризик порушення цілісності програмного забезпечення.

Зростання вішингу, діпфейків та обману особи

Злиття голосової соціальної інженерії та штучного інтелекту має визначити нову еру атак у 2026 році, зі значним зростанням прогнозованих атак Vishing (голосовий фішинг) з використанням технології DeepFake.

У другій половині 2025 року спостерігалось помітне зростання кількості атак Vish з використанням таких платформ, як Microsoft Teams. Впровадження таких функцій, як «Чат з будь-ким» у Teams, що дозволяє контактувати між людьми з різних орендарів, створює новий критичний вектор атаки. Зловмисник може легко видати себе за довірену особу, таку як ІТ-співробітник, щоб ініціювати атаку соціальної інженерії.

Послідовність атаки, яка стає небезпечно ефективною завдяки компонентам DeepFake та стиснутому часу виконання, зазвичай виглядає приблизно так:

1. Початковий контакт: Зловмисник зв'язується зі співробітником через платформу для співпраці та надсилає шкідливу URL-адресу.

2. Крадіжка облікових даних: Посилання запитує облікові дані, а потім намагається встановити інструмент RMM – знову ж таки, перетворюючи легітимне програмне забезпечення на зброю.

3. Латеральний рух та обман: У разі успіху зловмисник отримує віддалений доступ. Щоб ще більше маніпулювати жертвою та її колегами, зловмисник впроваджує технологію DeepFake. Ця дуже переконлива імперсонація може бути використана для авторизації шахрайських транзакцій, отримання подальшого доступу або створення хибного відчуття безпеки.

Очікується, що це поєднання Vishing та DeepFake значно скоротить загальний час атаки та прискорить витік даних і фінансову крадіжку, швидко забезпечуючи горизонтальне переміщення та долаючи захисні механізми перевірки людиною.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Таргетинг SAAS

SaaS, на який спрямовані атаки активних неорганізованих злочинних груп (таких як Scattered Spider), продовжаться у 2026 році, починаючи з атак соціальної інженерії та спрямованих на клієнтські SaaS-додатки, захищені застосунками єдиного входу.

АРТ

АРТ або групи національних держав, що націлюються на них, продовжуватимуть викликати серйозне занепокоєння у 2026 році з різних причин, таких як атаки на інтелектуальну власність з певних галузей промисловості, таких як напівпровідникова промисловість, або під час геополітичної напруженості. Ми прогнозуємо, що групи загроз з Близького Сходу, Південно-Східної Азії та Росії продовжуватимуть атакувати регіони, що їх цікавлять.

Ефективність у зловмисному вчинку: збільшення повторного використання коду та ребрендинг

У 2026 році розробники шкідливого програмного забезпечення надаватимуть пріоритет ефективності, що призведе до помітного збільшення повторного використання коду для розробки та ребрендингу нових шкідливих інструментів. Ця стратегія дозволяє зловмисникам швидко виконувати ітерації, запускати нові кампанії з мінімальним часом розробки та уникати статичних сигнатур виявлення.

Яскравим прикладом цього є еволюція бразильського банківського шкідливого програмного забезпечення:

– Еволюція Coyote: У 2024 році повідомлялося, що шкідливе програмне забезпечення Coyote було спрямоване на приблизно 60 бразильських вебсайтів. На початку 2025 року дослідники CyberProof помітили оновлення, яке розширило його таргетинг до понад 1000 вебсайтів.

– Зв'язок Maverick: Пізніше у 2025 році поява банківського шкідливого програмного забезпечення Maverick продемонструвала значну схожість з Coyote.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Ці спільні характеристики варіювалися від початкового стилю зараження (часто з використанням WhatsApp) до ідентичних механізмів шифрування.

Ця закономірність свідчить про те, що зрілі, усталені кодові бази систематично використовуються повторно, часто з незначними модифікаціями, для створення «нових» загроз. Такий швидкий ребрендинг ставить перед командами безпеки завдання не відставати, оскільки вони повинні постійно адаптуватися до функціонально схожих, але зовні відмінних сімейств шкідливих програм.

Прогнози на 2026 рік вказують на критичну потребу для організацій у пріоритезації проактивних стратегій захисту, які виходять за рамки традиційного виявлення на основі сигнатур. Увага має зміститися на поведінкову аналітику для виявлення зловживань легітимними інструментами, надійні протоколи перевірки особи для протидії передовій соціальній інженерії та всебічне розуміння еволюції коду зловмисників для передбачення нових варіантів шкідливого програмного забезпечення.

Майбутнє безпеки залежатиме від складного захисту, який зможе розрізнити зловмисні наміри від законної цифрової активності.

Ідентичність замінює периметр, всюди

З усіх пріоритетів, які мають CISO на 2026 рік, ідентичність виділяється як та, що стоїть на першому плані.

Традиційний периметр розчинився та був замінений складною мережею користувачів-людей, облікових записів сервісів, API, інтеграцій SaaS та зростаючої кількості агентів штучного інтелекту. Кожна нова програма, робочий процес або автоматизація вводять нові ідентифікаційні дані та нові довірчі відносини, часто швидше, ніж моделі управління та відповідності можуть адаптуватися.

Крадіжка облікових даних, підвищення привілеїв та використання довірчих відносин стали найнадійнішими точками входу для зловмисників. Але проблема виходить за рамки крадіжки: величезна кількість ідентифікаційних

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

даних, особливо нелюдських, ускладнює видимість та контроль навіть для найобізнаніших команд безпеки.

Ідентифікація більше не є лише однією з областей безпеки. Це сполучний рівень, який об'єднує доступ до хмари, поведінку мережі, активність кінцевих точок та інтеграції третіх сторін. На практиці безпека ідентифікації у 2026 році не є додатковою функцією. Це площина керування.

Хмарні середовища

Хмарні середовища змінили те, як організації будують та масштабуються. Вони також фундаментально змінили те, як поширюються ризики та атаки.

Неправильні конфігурації залишаються однією з найпоширеніших причин впливу хмарних технологій, але глибша проблема полягає в швидкості. Інфраструктура змінюється швидше, ніж політики, дозволи та моніторинг можуть встигати за цим. Вікно для помилок ще більше розширюється, коли дії за допомогою штучного інтелекту входять до автоматизованих розгортань, скриптів або інтеграцій.

Водночас підприємства стають дедалі залежнішими від SaaS-платформ, постачальників та екосистем на основі API. Атаки на ланцюги поставок більше не є крайніми випадками; вони є системними за своєю природою. Компрометація не завжди відбувається через шкідливе програмне забезпечення чи експлойти, а через легітимні шляхи доступу, яким команди безпеки неявно довіряють.

До 2026 року хмарні ризики та ризики третіх сторін зливаються в одну реальність: безперервність бізнесу. Сьогодні вплив порушення вимірюється не лише втратою даних, але й з точки зору операційних збоїв, впливу на регуляторні органи та втрати довіри.

Що це означає для керівників інформаційних систем у 2026 році

Реактивні операційні моделі ледве справляються з величезним обсягом сповіщень, фрагментованим інструментарієм та ручним розслідуванням. Розрізнена видимість між ідентифікаторами, хмарою, мережею та третіми сторонами створює сліпі зони, які зловмисники легко використовують.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Водночас, ради директорів та керівники очікують чіткіших відповідей щодо ризиків, стійкості та рентабельності інвестицій у безпеку.

Тенденції автоматизації безпеки не про заміну людського досвіду, а про його забезпечення. Співвідношення сигналів між рівнями, визначення пріоритетів того, що дійсно важливо, та надання командам можливості діяти швидко та впевнено. CISO еволюціонує до ролі оркестрації: узгодження технологій, процесів та людей навколо спільного розуміння ризику.

У цьому контексті кібербезпека у 2026 році полягає не стільки у запобіганні кожному інциденту, скільки у забезпеченні того, щоб організація могла адаптуватися до інцидентів та продовжувати їх роботу у разі їх виникнення.

Визнання цих тенденцій – це лише перший крок. Для виконання потрібна операційна модель безпеки, яка відображає, як сучасні атаки розгортаються на різних рівнях, ідентифікаторах та екосистемах.

Організації, що орієнтовані на майбутнє, вже відходять від ізольованих захисних механізмів до інтегрованих підходів, що об'єднують виявлення, розвідку, автоматизацію та реагування. Вони переосмислюють успіх не через відсутність інцидентів, а через вимірювану стійкість: зменшення впливу, швидше відновлення та чіткіша видимість на рівні керівництва.

Це стратегічний порядок денний, який формує майбутнє забезпечення безпеки корпоративної мережі від зловмисних дій. Більше того, рішення, які прийматимуть керівники інформаційних систем сьогодні, визначатимуть, наскільки готові їхні організації до того, що буде далі.

3.2 Розробка структурної схеми

На зорі цифрового зв'язку захист інформації в Інтернеті спирався на просту ідею: обидві сторони використовували один і той самий секретний ключ для шифрування та розшифрування повідомлень. Поки цей ключ залишався приватним, дані залишалися в безпеці. Цей підхід, відомий як симетрична

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

криптографія, ліг у основу раннього безпечного зв'язку.

Навіть попри ефективне шифрування, довіра стала новим і значним викликом. Безпечне використання секретного ключа ставало все складнішим, оскільки мережі зростали, і для обміну даними потрібно було більше систем. Один перехоплений ключ міг поставити під загрозу всі захищені повідомлення. Зі збільшенням кількості користувачів, сервісів і машин керування спільними секретами стало складним і крихким, що ускладнило масштабування безпечного зв'язку.

Криптографія з відкритим ключем виникла для вирішення цієї проблеми шляхом розділення шифрування та дешифрування на два математично пов'язані ключі. Один ключ можна було використовувати відкрито, тоді як інший залишався приватним та суворо контрольованим. Дані, зашифровані відкритим ключем, можна було розшифрувати лише за допомогою відповідного закритого ключа, що усуває необхідність попереднього обміну секретами. Такий підхід забезпечує безпечний зв'язок через ненадійні мережі та робить можливими такі технології, як HTTPS, цифрові підписи та безпечна автентифікація.

Сьогодні, хоча асиметрична криптографія добре вивчена, безпечне управління цими ключами в сучасній інфраструктурі є причиною більшості проблем безпеки, особливо коли організації починають готуватися до постквантової криптографії. У цьому розділі ми розглянемо найкращі практики ефективного управління відкритими та закритими ключами у 2026 році.

Розуміння відкритих та закритих ключів

Щоб ефективно керувати ключами, важливо спочатку зрозуміти, що таке відкритий та закритий ключі, а також яку роль вони відіграють у безпечному зв'язку. Ці два компоненти асиметричної криптографії математично пов'язані, але вони виконують дуже різні функції. Розуміння того, як функціонує кожен ключ і чому він важливий, закладає основу для правильного керування ними.

Що таке закритий ключ?

Закритий ключ – це конфіденційна частина асиметричної пари ключів. Він повинен завжди залишатися секретним і ніколи не повинен бути переданий

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

третім особам. Хоча він математично пов'язаний з відповідним відкритим ключем, отримання закритого ключа з відкритого ключа є обчислювально нездійсненним за допомогою сучасної криптографії.

На практиці володіння закритим ключем слугує криптографічним доказом ідентифікації. Коли він використовується для розшифрування даних або створення цифрового підпису, він надає математичний доказ того, що операцію виконав законний власник ключа. Будь-хто, хто отримує доступ до закритого ключа, фактично отримує можливість видавати себе за його власника. Саме тому закриті ключі вважаються цінними активами безпеки та потребують найсильнішого захисту.

Що таке відкритий ключ?

Відкритий ключ – це спільний аналог закритого ключа. Він призначений для вільного розповсюдження без шкоди для безпеки. Будь-хто може використовувати відкритий ключ для шифрування даних, призначених для власника ключа, або для перевірки цифрових підписів, створених за допомогою відповідного закритого ключа.

Безпека цієї моделі базується на односторонньому математичному зв'язку між ключами. Дані, зашифровані відкритим ключем, можна розшифрувати лише за допомогою відповідного закритого ключа, а підписи, створені закритим ключем, можна перевірити лише за допомогою відповідного відкритого ключа. Це дозволяє встановити довіру без попереднього обміну секретом.

Тепер, коли ми розуміємо, що таке публічні та закриті ключі, давайте коротко розглянемо, як вони працюють разом у реальному світі.

Як разом використовуються відкриті та закриті ключі?

Відкритий та закритий ключі не є конкурентами; вони працюють як узгоджена пара. Кожен ключ виконує певну роль, і разом вони забезпечують безпечний зв'язок, автентифікацію та захист даних.

Коли хтось хоче надіслати захищену інформацію, він використовує відкритий ключ одержувача. Це блокує дані таким чином, що їх може розблокувати лише відповідний закритий ключ. Оскільки відкритий ключ не

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

містить секретів, ним можна вільно ділитися.

Потім власник використовує закритий ключ для розблокування даних або підтвердження особи. Коли закритий ключ створює цифровий підпис, будь-хто, хто має відповідний відкритий ключ, може перевірити, чи справді повідомлення надійшло з очікуваного джерела та не було змінено.

Цей чіткий розподіл обов'язків робить асиметричну криптографію настільки потужною. Відкриті ключі забезпечують безпечний обмін та верифікацію, тоді як закриті ключі встановлюють ідентифікацію та контроль. Разом вони дозволяють системам обмінюватися даними, автентифікувати користувачів та будувати довіру, не розголошуючи секретну інформацію заздалегідь.

Як постквантова криптографія впливає на управління ключами?

Постквантова криптографія (PQC) впроваджує нові криптографічні алгоритми, розроблені для захисту від атак квантових комп'ютерів. Хоча ці алгоритми зберігають ті ж цілі безпеки, що й класична криптографія з відкритим ключем, вони суттєво змінюють те, як ключі генеруються, зберігаються, розподіляються та керуються протягом їхнього життєвого циклу. Отже, PQC фундаментально впливає на практику управління ключами. Нижче наведено найважливіші наслідки:

1. Великі ключі та артефакти

Більшість стандартизованих алгоритмів PQC використовують набагато більші розміри ключів, ніж класичні алгоритми, такі як RSA або ECC. Відкриті ключі, закриті ключі та підписи можуть бути в кілька разів більшими, що впливає на:

- Розміри сертифікатів та накладні витрати на передачу.
- Вимоги до зберігання в сховищах ключів та HSM.
- Затримка мережі під час встановлення зв'язку (наприклад, TLS).
- Системи ведення журналу, резервного копіювання та архівування.

Системи керування ключами повинні бути перевірені для обробки великих криптографічних об'єктів без зниження продуктивності або неочікуваних

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

збоїв.

2. Продуктивність та вплив на операційну діяльність

Деякі алгоритми PQS створюють підвищені обчислювальні витрати, особливо для генерації ключів та перевірки підпису. Хоча це прийнятно для багатьох випадків використання, це може вплинути на:

- Системи автентифікації великих обсягів.
- Конвеєри підписання коду.
- Центри сертифікації, що видають велику кількість сертифікатів.
- Середовища з обмеженими ресурсами, такі як Інтернет речей або периферійні пристрої.

У 2026 році найкращою практикою є раннє вимірювання та моделювання впливу на продуктивність, що забезпечує надійне масштабування ключових операцій життєвого циклу (випуск, ротація, перевірка) під робочими навантаженнями PQS.

3. Гібридні криптографічні моделі

Галузь переходить до використання гібридної криптографії, де класичні та постквантові алгоритми використовуються разом. У гібридних моделях:

- Ключі та сертифікати можуть включати як класичні, так і PQS алгоритми.
- Довіра зберігається доти, доки хоча б один алгоритм залишається безпечним.
- Ризик міграції знижується без шкоди для сумісності.

Системи управління ключами повинні підтримувати:

- Кілька алгоритмів на ключ або сертифікат.
- Стратегії паралельного обертання.
- Чітке забезпечення дотримання політики щодо гібридного використання.

Це збільшує складність і посилює потребу в централізованому, автоматизованому управлінні ключами.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

4. Довший перехід, довша експозиція

Квантові атаки, такі як Harvest-Now, Decrypt-Later (HNDL), спрямовані на дані, зашифровані сьогодні, але розшифровані в майбутньому. Це означає:

- Довговічні ключі шифрування та сертифікати, що захищають довгострокові конфіденційні дані, перебувають під високим ризиком.
- Архівні зашифровані дані вже можуть бути вразливими.
- Ротація ключів та криптографічна гнучкість більше не є необов'язковими.

Організаціям слід скоротити термін служби ключів, надати пріоритет алгоритмам, безпечним для PQS, для довгострокової конфіденційності та підтримувати інвентаризацію, яка ідентифікує квантово-вразливі ключі.

5. Криптографічна гнучкість як основна вимога

Стандарти PQS продовжуватимуть розвиватися, і деякі алгоритми можуть бути застарілими або вдосконаленими з часом. Платформи керування ключами повинні підтримувати:

- Гнучкість алгоритмів без переписування застосунків.
- Вибір криптографічних примітивів на основі політик.
- Безперебійна ротація між алгоритмами.
- Співіснування застарілих, гібридних та постквантових ключів.

Жорстким системам, які використовують один алгоритм або фіксовану структуру ключів, буде важко адаптуватися.

Оскільки організації впроваджують нові криптографічні алгоритми та рухаються до постквантової готовності, ландшафт управління ключами стає складнішим, ніж будь-коли. Ключі більше не є статичними активами; вони стають більшими, обертаються частіше та повинні співіснувати в гібридних та багатохмарних середовищах.

Цей зсув порушує важливе питання: якщо сучасна криптографія вже дає збій через неправильно керовані ключі, що станеться, коли розміри ключів зростуть, життєві цикли скоротяться, а алгоритми змінюватимуться частіше? Щоб відповісти на це питання, ми повинні спочатку зрозуміти, чому управління

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

ключами має значення, і які помилки найчастіше допускають організації. Без надійних практик управління ключами підвищена складність, що вноситься PQS, лише посилює існуючі вразливості.

Чому важливе управління ключами та найпоширеніші помилки?

Навіть найсильніше шифрування не працює, якщо ключі неправильно керуються. Приватні ключі формують основу довіри та ідентифікації в цифрових системах. Неправильне використання ключів призводить не лише до послаблення безпеки, а й до реальних наслідків, таких як несанкціонований доступ, витоки даних, перебої в обслуговуванні та втрата цілісності системи. Нижче наведено деякі з найпоширеніших помилок в управлінні ключами, які допускають організації, кожна з яких демонструє, як погане поводження з ключами може послабити безпеку та довіру.

Жорстко закодовані клавіші

Вбудовування ключів безпосередньо в код програми, файли конфігурації або скрипти автоматизації робить їх доступними для будь-кого, хто має доступ до вихідного коду. GitHub повідомив, що лише у 2024 році в публічних репозиторіях GitHub було виявлено понад 39 мільйонів витіків секретних даних, включаючи ключі API, облікові дані, токени та інші конфіденційні дані. Після виявлення ці ключі дозволяють зловмисникам повністю обійти шифрування або автентифікацію.

Надмірно поширені приватні ключі

Приватні ключі призначені для зберігання однією особою. Їх спільне використання кількома користувачами, командами або системами швидко збільшує поверхню ризику. Кожна додаткова копія збільшує ймовірність випадкового розкриття, неправильного використання або крадіжки, що ускладнює визначення відповідальності або обмеження компрометації.

Повторне використання ключа

Повторне використання одного ключа для шифрування, підпису та автентифікації збільшує ризик. Якщо цей ключ скомпрометовано, одночасно виходять з ладу кілька засобів контролю безпеки. Таке повторне використання

ключа перетворює один інцидент на каскадний збій у всіх системах і службах.

Слабка генерація ключів

Ключі, згенеровані з використанням недостатньої ентропії або застарілих генераторів випадкових чисел, можуть бути передбачуваними. Відомим прикладом є рання вразливість Debian OpenSSL, де низька випадковість робила криптографічні ключі тривіальними для грубої сили. Слабкі ключі підривають безпеку в момент створення та залишаються вразливими протягом усього свого життєвого циклу.

Некеровані або осиротілі ключі

Ключі, які залишаються активними ще довго після звільнення працівника, зміни ролі або виведення систем з експлуатації, становлять серйозні ризики для безпеки. Ці осиротілі ключі непомітно накопичуються та призводять до їх розповсюдження, і часто залишаються непоміченими, оскільки їх рідко контролюють. В результаті вони стають легкою мішенню для зломисників або джерелом випадкового неправильного використання.

Неможливість скасування або ротації ключів

Ключі, які ніколи не ротуються, не мають терміну дії або не відкликаються, залишаються довіреними на невизначений термін. Зломисники можуть використовувати старі або скомпрометовані ключі, які системи все ще приймають як дійсні, перетворюючи історичні помилки на сучасні порушення.

Ручне керування ключами

Покладання на ручні процеси створення, розповсюдження, ротації або відкликання ключів не масштабується в сучасних середовищах. Пропущені терміни дії, неправильні дозволи та забуті відкликання є поширеними явищами, які часто залишаються непоміченими, доки не спричинять збої або інциденти безпеки. Оскільки системи стають більш динамічними та розподіленими, ручне керування ключами стає прямим ризиком для надійності, безпеки та відповідності вимогам.

Неправильно налаштовані центри сертифікації або каталоги ключів

Відкриті ключі визначають довіру на основі того, хто їх видав і де ця

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

довіра закріплена. Якщо центри сертифікації (ЦС) або каталоги ключів налаштовані неправильно, системи можуть довіряти неправильним ключам або блокувати легітимні. Наприклад, якщо система все ще довіряє ЦС, який мав бути видалений, зловмисник може пред'явити підроблений сертифікат, який виглядає дійсним. І навпаки, якщо легітимний ЦС або ключ випадково видалено або неправильно налаштовано, дійсні сертифікати можуть бути відхилені, що призведе до збою захищених служб або блокування доступу користувачів до ресурсів.

Відсутність моніторингу використання ключів

Багато організацій не відстежують, коли, де і як використовуються ключі. Без видимості незвичайна або зловмисна активність може залишатися непоміченою протягом тривалого часу, що збільшує як вплив, так і вартість інцидентів.

Ці пастки показують, що управління ключами не є необов'язковим. Це основа безпечних криптографічних операцій. Незалежно від того, наскільки сильними є ваші алгоритми шифрування, погане поводження з ключами може звести нанівець усі захисти. У сучасних ІТ-середовищах, з хмарними сервісами, автоматизованими конвеєрами та розподіленими командами, дисципліноване управління ключами є важливим для забезпечення безпеки, відповідності та довіри у великих масштабах.

Найкращі практики управління закритими ключами

Після вивчення поширених помилок стає зрозуміло, що закриті ключі є одними з найчутливіших та найважливіших активів у будь-якій криптографічній системі. Їх захист не підлягає обговоренню, оскільки після компрометації закритого ключа зловмисники можуть видавати себе за інших користувачів, розшифровувати конфіденційні дані або повністю обходити автентифікацію. Ось як організації можуть ефективно керувати ними у 2026 році:

1. Генерація безпечних ключів

Приватні ключі слід генерувати в безпечних середовищах, таких як апаратні модулі безпеки (HSM) або модулі довіреної платформи (TPM).

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

Генерація ключів у захищеному середовищі зменшує ризики та запобігає перехопленню або неправильному поводженню з ключами під час створення чи імпорту. Крім того, генеруйте ключі з достатньою довжиною та надійністю ключа (наприклад, RSA-3072 або вище) та уникайте жорсткого кодування ключів у вашому вихідному коді.

2. Використання ключів за призначенням

Криптографічні ключі повинні бути суворо обмежені однією метою. Ключі шифрування, ключі підпису, ключі автентифікації та ключі центру сертифікації виконують принципово різні ролі та ніколи не повинні використовуватися взаємозамінно.

Наприклад, ключ, який використовується для підписання програмного забезпечення або сертифікатів, ніколи не повинен використовуватися для шифрування даних. Так само закриті ключі TLS не слід повторно використовувати для підписання документів або автентифікації поза їхнім цільовим контекстом. Розділення ключів за призначенням обмежує радіус вибуху будь-якої компрометації.

3. Безпечне сховище

Забезпечте, щоб закриті ключі ніколи не потрапляли до чужих рук. Використовуйте апаратні модулі безпеки (HSM), що відповідають стандарту FIPS 140-3, або безпечні сховища ключів, такі як AWS KMS або Azure Key Vault. Ключі ніколи не повинні зберігатися у відкритому тексті, файлах конфігурації або незахищеному сховищі. Апаратний захист гарантує, що ключі не можна витягти, якщо їх налаштовано як неекспортовані, навіть привілейованими користувачами.

4. Обмежити доступ

Контроль доступу йде пліч-о-пліч із безпечним зберіганням ключів. Хоча ключі повинні бути доступні для підтримки операцій, вони повинні бути доступні лише авторизованим користувачам і системам. Обмежте доступ суворо тими, кому він потрібен, забезпечуючи контроль доступу на основі ролей, принцип найменших привілеїв та багатофакторну автентифікацію для використання

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

ключів.

5. Розподіл обов'язків

Жоден окремий користувач чи система не повинні контролювати весь життєвий цикл ключа. Розділення обов'язків, таких як генерація, затвердження, розгортання та скасування ключів, знижує ризик внутрішніх загроз та випадкового неправомірного використання. Цей контроль особливо важливий у регульованих середовищах або середовищах з високим рівнем гарантування.

6. Ключове зберігання та нагляд

Призначте спеціальних ключових відповідальних за управління та нагляд, а не за безпосереднє операційне використання. Відповідальні затверджують доступ, забезпечують дотримання політики, контролюють використання та забезпечують своєчасну ротацію та скасування. Таке розділення контролю та виконання покращує підзвітність та зменшує ризик мовчазного використання.

7. Ключова сегментація на основі ризику

Класифікуйте ключі на основі чутливості, щоб забезпечити найвищий рівень захисту цінних криптографічних активів, тоді як ключі з меншим ризиком можуть мати стандартні елементи керування, що гарантує зосередження ресурсів там, де вони найбільш важливі.

8. Ротація та анулювання, керовані життєвим циклом

NIST рекомендує визначати терміни життя ключів на основі алгоритму, використання та профілю ризику, а не фіксованих часових інтервалів. Ключі підпису з великим обсягом даних або ключі, що розкриваються ззовні, можуть вимагати частішої ротації. Скомпрометовані або застарілі ключі необхідно негайно відкликати. Крім того, автоматизація зменшує людські помилки та забезпечує послідовне дотримання політик ротації та відкликання.

9. Безпечне резервне копіювання та відновлення

Втрата закритого ключа може мати катастрофічні наслідки, особливо для зашифрованих даних, які неможливо відновити без нього. Зберігайте зашифровані резервні копії в окремих, безпечних географічних місцях із суворим контролем доступу. Процедури відновлення слід періодично тестувати, щоб

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

забезпечити швидке та безпечне відновлення ключів за потреби.

10. Безперервний моніторинг та аудит

Прозорість використання ключів є критично важливою. Організації повинні постійно контролювати, які ключі використовуються, коли до них здійснюється доступ і ким. Централізоване ведення журналу, аудит та виявлення аномалій допомагають виявити неправомірне використання, скомпрометовані ключі або помилки конфігурації на ранній стадії. Ведення повного та актуального інвентаризації ключів додатково підтримує реагування на інциденти, дотримання вимог та управління.

11. План реагування на інциденти

Незважаючи на всі запобіжні заходи, компрометація може статися. Встановіть чіткі процедури реагування на інциденти для анулювання, повторного видання, відновлення послуг та повідомлення зацікавлених сторін. Підготовка мінімізує збитки та простої.

12. Комплексна автоматизація

Автоматизуйте операції життєвого циклу ключів, включаючи створення, ротацію, скасування та резервне копіювання. Автоматизація зменшує залежність від ручних процесів та забезпечує послідовне дотримання політик.

Дотримання цих найкращих практик гарантує, що закриті ключі використовуються виключно за призначенням, залишаються захищеними протягом усього їхнього життєвого циклу та продовжують служити надійною основою для безпечного шифрування та автентифікації. Після належного захисту закритих ключів наступним кроком є забезпечення такого ж ретельного розподілу, довіри та управління відкритими ключами.

Найкращі практики управління відкритими ключами

Відкриті ключі дозволяють системам перевіряти особи та встановлювати довіру без необхідності попереднього обміну секретними даними, що робить їх належне управління важливим для безпечної роботи. Хоча вони розроблені для спільного використання, неправильне поводження з відкритими ключами може призвести до вразливостей, порушення довірчих відносин та створення

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

операційних ризиків, особливо у великих та розподілених середовищах. Керування відкритими ключами з тією ж дисципліною, що й закритими ключами, є важливим для підтримки безпеки та надійності систем. Ось як ефективно керувати відкритими ключами у 2026 році:

1. Перевірка та перевірка відкритих ключів

Відкриті ключі можуть поширюватися відкрито, але їхня автентичність має бути перевірена за допомогою надійних механізмів, таких як сертифікати, відбитки пальців або точки довіри. Це включає перевірку ланцюжків сертифікатів, довірених корневих сертифікатів та політик видачі. Довіряти слід лише ключам, які походять від затверджених центрів сертифікації та відповідають криптографічним стандартам організації.

Оскільки впроваджуються сертифікати PQS та гібридні сертифікати, логіку перевірки необхідно оновлювати, щоб правильно інтерпретувати нові алгоритми, розміри підписів та структури сертифікатів, аби уникнути хибної довіри або неочікуваних збоїв.

2. Централізована видимість та інвентаризація

Організації повинні вести централізований облік усіх ключів та сертифікатів, що використовуються. Цей облік повинен чітко показувати, які системи довіряють яким ключам і для якої мети. Без видимості накопичуються невикористані або застарілі ключі, що збільшує ризик неправильної конфігурації, розповсюдження ключів та несанкціонованого доступу.

3. Закінчення терміну дії та анулювання монітора

Сертифікати відкритого ключа повинні мати чітко визначені терміни дії та механізми відкликання. Системи повинні постійно перевіряти їхню дійсність, терміни дії та статус відкликання за допомогою таких методів, як CRL або OCSP. Ключам, термін дії яких минув або які відкликані, ніколи не слід довіряти, що допомагає зменшити ризик зіткнення зі скомпрометованими або застарілими ключами.

4. Короткотривалі сертифікати

Використовуйте короткочасні сертифікати відкритих ключів, коли це

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

можливо, щоб обмежити ризики. Коротший термін дії зменшує вплив скомпрометованих ключів і мінімізує залежність від повільних механізмів відкликання.

Галузеві тенденції вже рухаються в цьому напрямку: термін дії сертифікатів TLS скоротився до менш ніж 398 днів, при цьому зростає впровадження коротших термінів дії, таких як 200 днів і 100 днів, а також пропонуються скоротити їх до 47 днів у найближчі роки. Впровадження короткострокових сертифікатів зараз допоможе організаціям випереджати ці зміни, одночасно зміцнюючи безпеку та операційну стійкість.

Коротший термін служби сертифікатів також зменшує вразливість під час постквантового переходу, обмежуючи ризик того, що квантово-вразливі відкриті ключі залишатимуться довіреними протягом тривалого часу.

5. Довірчі відносини в аудиті

Довірчі відносини слід періодично переглядати, щоб переконатися, що вони залишаються навмисними та точними. Застарілі точки довіри, неправильно налаштовані ієрархії сертифікатів або неавторизовані ключі можуть дозволити зловмисникам видавати себе за інші служби або користувачів. Регулярні аудити допомагають виявляти та усувати ці ризики до того, як вони будуть використані.

6. Інтеграція з DevOps та хмарою

Вбудовуйте управління відкритими ключами та сертифікатами в конвеєри CI/CD, хмарні сервіси та інтеграції API. Автоматизована видача та розгортання гарантують, що нові робочі навантаження, контейнери та безсерверні функції отримуватимуть правильні сертифікати та довірені відкриті ключі без ручного втручання. Забезпечте контроль на основі політик щодо видачі, розгортання та довірчих якорів сертифікатів, ведіть централізований журнал використання та перевірки сертифікатів, а також постійно контролюйте довірчі відносини, щоб забезпечити безпеку та відповідність вимогам у динамічних хмарних середовищах.

Ефективне управління відкритими ключами забезпечує правильне встановлення довіри, її підтримку з часом та безпечне масштабування в різних

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

системах. Коли як відкриті, так і закриті ключі управляються дисципліновано та прозоро, організації краще підготовлені до вирішення ширших проблем сучасної, автоматизованої та постійно мінливої інфраструктури.

Індивідуальні послуги шифрування

Сучасні міркування щодо ключового менеджменту у 2026 році

Ефективне управління ключами є важливим для захисту сучасних цифрових систем. З розвитком технологій розвиваються і стратегії та інструменти, необхідні для захисту криптографічних ключів протягом усього їхнього життєвого циклу. Організації впроваджують сучасні підходи, що зосереджені на автоматизації, гнучкості та прозорості. Ключові міркування включають:

1. Криптографічна гнучкість та постквантова готовність

Системи, побудовані на основі фіксованих алгоритмів або жорстко закодованих криптографічних припущень, важко адаптувати до розвитку стандартів . З огляду на те, що постквантова криптографія вже на горизонті, організації повинні розробляти системи керування ключами, які підтримують гнучкість алгоритмів. Це включає можливість впровадження нових алгоритмів, безперешкодної ротації ключів та запуску гібридних криптографічних моделей без перебоїв у роботі.

2. Автоматизоване управління життєвим циклом ключів

Використовуйте централізовані та автоматизовані платформи управління життєвим циклом ключів для створення, ротації, відкликання та видалення ключів. Методи ручного відстеження, такі як електронні таблиці або спеціальні сценарії, не масштабуються та часто призводять до втрати ключів, пропущених ротацій та непослідовного застосування політик. Автоматизоване управління життєвим циклом зменшує людські помилки, послідовно застосовує криптографічну політику та гарантує, що ключі ніколи не використовуються поза межами їхньої передбачуваної області застосування або терміну служби.

3. Хмарні та тимчасові робочі навантаження

Сучасні хмарні середовища залежать від контейнерів, безсерверних

функцій та динамічної інфраструктури, які швидко створюються та завершуються. У таких ефемерних робочих навантаженнях криптографічні ключі часто повинні існувати лише тимчасово та повинні автоматично надаватися та ротуватися. Використання короткочасних ключів мінімізує довгостроковий вплив та гарантує, що безпека відповідає вимогам високодинамічних систем.

4. Інтеграція з нульовою довірою та DevSecOps

Ключі все частіше вбудовуються в робочі процеси, керовані політиками, а не розподіляються вручну. Архітектури нульової довіри вимагають постійної перевірки, а це означає, що доступ до ключів має динамічно оцінюватися на основі ідентифікації, контексту та політики. Інтеграція управління ключами в конвеєри CI/CD та робочі процеси DevSecOps гарантує безпечне надання ключів, реєстрацію та моніторинг доступу, а також раннє виявлення неправильного використання без уповільнення швидкості розробки.

5. Моніторинг та автоматизація в режимі реального часу

Сучасне управління ключами вимагає перегляду використання, стану та відповідності ключів у режимі реального часу. Організації повинні збирати детальну телеметрію щодо того, коли здійснюється доступ до ключів, як вони використовуються та якими ідентифікаторами. Автоматизовані сповіщення, інформаційні панелі та забезпечення дотримання політик допомагають виявляти аномальну поведінку, несанкціоноване використання та відхилення конфігурації, перш ніж вони переростуть у інциденти безпеки.

6. Довіра, підкріплена апаратним забезпеченням

Оскільки захист, що базується лише на програмному забезпеченні, стає легше обійти, апаратний захист ключів відіграє вирішальну роль. Використання HSM або TPM, що відповідають стандарту FIPS 140-3, створює міцну основу довіри, де закриті ключі ніколи не залишають межі захищеного обладнання. Це особливо важливо для кореневих та видаючих ключів СА, ключів підпису коду та ключів автентифікації з високим рівнем надійності. Застосовуйте політики, які запобігають вилученню ключів, вимагають апаратно-підтримуваних операцій для підписання та розшифрування, а також підтримують надійне журналювання аудиту.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

7. Міжсередовищева та багатохмарна узгодженість

Організації, що працюють у локальних, хмарних та багатохмарних середовищах, повинні застосовувати єдині політики управління ключами незалежно від того, де використовуються ключі. Це включає стандартизовані алгоритми, терміни дії ключів, правила ротації, засоби контролю доступу та вимоги до аудиту. Централізовані платформи управління повинні керувати політиками та видимістю, водночас забезпечуючи гнучкість моделей розгортання, щоб запобігти прогалинам у безпеці, дрейфу та непослідовним моделям довіри.

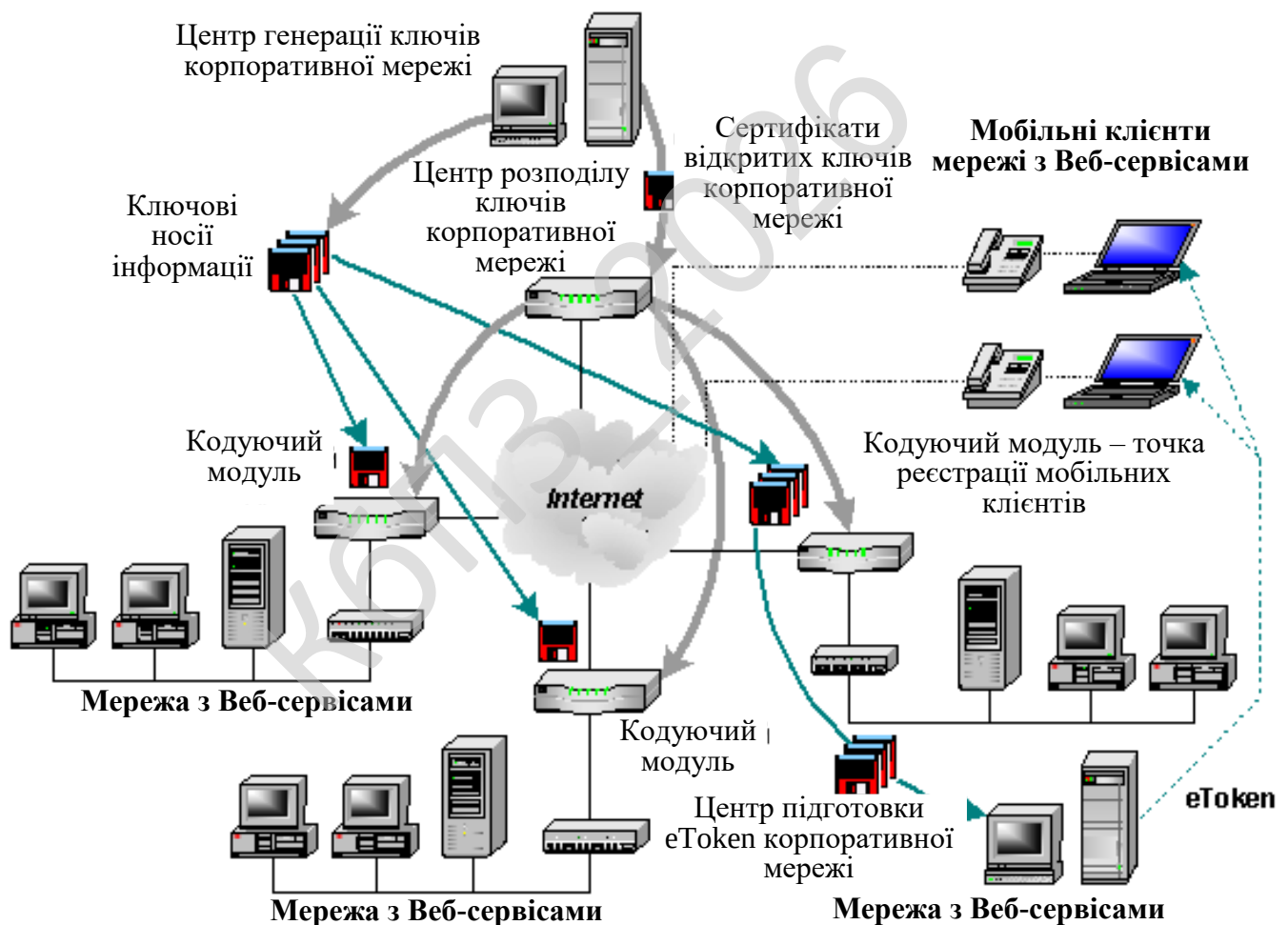


Рисунок 3.1 – Структурна схема системи

8. Відповідність, аудитність та узгодженість нормативних актів

Нормативні акти дедалі більше вимагають демонстративного контролю над криптографічними ключами. Системи управління ключами повинні підтримувати журнали аудиту, журнали доступу, докази політики та автоматизовану звітність для відповідності вимогам, таким як NIST, HIPAA, PCI DSS та інші регіональні нормативні акти.

Сучасне управління ключами у 2026 році виходить за рамки захисту секретів. Воно вимагає автоматизації, прозорості та адаптивності, щоб йти в ногу з мінливою інфраструктурою та ландшафтом загроз. Організації, які розробляють управління ключами з урахуванням гнучкості та спостережливості в його основі, краще оснащені для підтримки безпеки, відповідності та довіри в масштабах.

3.3 Розробка функціональної схеми

Розглянемо функціональну схему розробленої системи. Вона зображена на рисунку 3.2. Як видно з рисунку система складається з наступних елементів:

1. Блок визначення алгоритму для формування ключа. Він може бути для симетричних та асиметричних алгоритмів. У першому випадку формується один ключ, який таємно розсилається учасникам зашифрованого обміну даними, у другому випадку генерується пара ключів: відкрий ключ – таємний ключ. Цей блок призначений для визначення по алгоритму шифрування параметрів та обмежень при формуванні ключа.

2. Генератор випадкових чисел. Призначений для генерації випадкових чисел, які використовуються як при генерації ключів шифрування/дешифрування, так й для формування іншої інформації потрібної для авторизації та автентифікації учасників інформаційного обміну.

3. Блок генерації ключів. Призначений для генерації ключів для учасників інформаційного обміну.

4. Блок формування сертифікатів. Призначений для формування

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

сертифікатів, які удостоверяють справжність ключів, та автентифікують користувачів.

5. База даних відкликаних сертифікатів, призначена для зберігання тих сертифікатів, які були зкомпроментовані.

6. Центр реєстрації.

Розглянемо ці блоки та їх функціональність більш детально.

Центр розподілу ключів

Центр розподілу ключів (ЦРК) призначений для обслуговування наступних запитів: на видання сертифікатів ЕЦП, на відкликання, призупинення й поновлення припиненої дії сертифікатів абонентів, сформованих на мережних вузлах або в Центрах Реєстрації для зовнішніх користувачів.

ЦРК забезпечує наступну функціональність:

1. Перше видання сертифіката підпису абонента відбувається в ЦРК разом з генерацією секретного ключа для нього. Подальше перевидання сертифіката може відбуватися як у ЦРК одночасно з формуванням нового секретного ключа (для завдань, що вимагають централізованої генерації й розподілу ключів), так і по запиту користувача корпоративної мережі, сформованого на його мережному вузлі.

2. Видання й реєстрація сертифікатів ЕЦП по запиту абонентів мережі. Запит на сертифікат являє собою шаблон сертифіката, що містить інформацію про абонента, його новий відкритий ключ підпису, передбачуваний термін дії сертифіката, а також інші параметри, що відповідають стандарту X.509. Запит може бути зареєстрований або автоматично або в результаті дій адміністратора ЦРК. Запит може бути відхилений. Після заповнення полів сертифіката сертифікат через Центр керування відправляється до користувача на комп'ютер.

3. Відкликання сертифікатів, призупинення дії сертифікатів, поновлення дії сертифікатів ЕЦП абонентів мережі. Ці дії виконуються адміністратором ЦРК. Довідник відкликаних сертифікатів розсилається абонентам мережі.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

4. Реєстрація довідників сертифікатів ЕЦП головних абонентів інших ЦРК. Після перегляду сертифікатів головних абонентів інших ЦРК і прийняття їх виробляється підпис такого довідника своїм головним абонентом (крос сертифікація). Завірений довідник розсилається по мережі у відповідності зі зв'язками своїх абонентів, і використовуються при перевірці сертифікатів ЕЦП абонентів інших ЦРК, що надіслали підписану інформацію на який-небудь вузол своєї мережі.

5. Аналогічним чином виконується імпорт, крос сертифікація й розсилання сертифікатів ЦРК інших виробників на основі сертифікованих СКЗІ. Документи, підписані з використанням зазначених СКЗІ, будуть перевірені тільки при наявності на комп'ютері сертифіката відповідного ЦРК, завіреного Головним абонентом свого ЦРК. По міркуваннях безпеки ЦРК вимагає крос сертифікації навіть для сертифікатів інших ЦРК, у ланцюжку сертифікатів яких утримується сертифікат, якому довіряє ЦРК.

6. Реєстрація довідників відкликаних сертифікатів ЕЦП із інших ЦРК. Такі довідники надходять із інших мереж автоматично, засвідчуються головним абонентом і розсилаються по мережі у відповідності зі зв'язками абонентів мережі. Імпорт довідників відкликаних сертифікатів зі ЦРК інших виробників не виробляється. Доступ до них здійснюється в процесі перевірки підпису по шляху, зазначеному в ЕЦП.

7. Обслуговування запитів зовнішніх користувачів. Зовнішній користувач реєструється на одному з пунктів реєстрації. Адміністратор створює запит на сертифікат ЕЦП для зовнішнього користувача й відсилає його в ЦРК для видання сертифіката. Запит на сертифікат перед відправленням у ЦРК підписується ключем підпису цього Адміністратора. Після введення в дію сертифіката зовнішній користувач зможе користуватися ним (підписувати документи) на будь-якому вузлі мережі із установленим ПЗ. Адміністратор може створювати запити на відкликання, призупинення дії, поновлення дії припиненого сертифіката ЕЦП зовнішніх користувачів.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

8. Видання й реєстрація сертифікатів ЕЦП для зовнішніх користувачів виконується тільки по запиту із Центра реєстрації (ЦР). Запит може бути зареєстрований або відхилений. Вторинні запити на сертифікати, якщо в них немає змін, і видача сертифікатів можуть оброблятися й видаватися автоматично. Сертифікати відправляється в ЦРК.

9. Відкликання сертифікатів, призупинення дії сертифікатів, поновлення дії сертифікатів ЕЦП зовнішніх користувачів може відбуватися по запиту зі ЦР, або самим Адміністратором ЦРК без запиту зі ЦР. Довідники відкликаних сертифікатів розсилаються по вузлах мережі.

10. Перегляд запитів і сертифікатів ЕЦП.

11. Розбір конфліктних ситуацій і експертизи правочинності й дійсності електронних документів, підписаних ЕЦП, виробляється відповідно до Документа "СКЗІ. Порядок розбору конфліктних ситуацій, пов'язаних із застосуванням ЕЦП".

12. Сервісні функції ЦРК.

ЦРК інформує адміністратора про витікання термінів дії різних сертифікатів за задане число днів до цього строку шляхом формування відповідних списків.

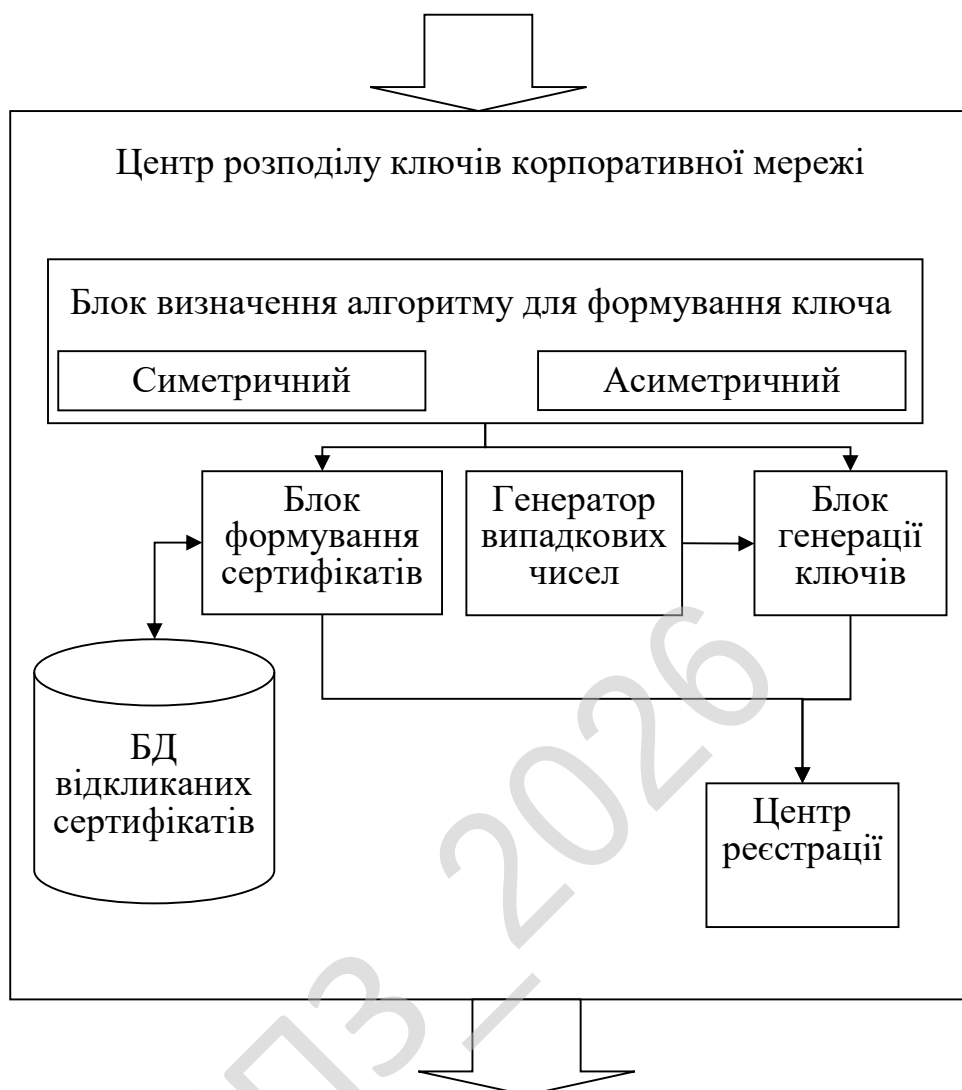
Автоматично формує архіви інформації через задані інтервали часу при наявності змін, що забезпечує можливість відновлення актуальної інформації.

Ключовий центр

Ключовий центр (КЦ) забезпечує захист інфраструктури ЕЦП за допомогою створення системи шифрування інформації у всіх процедурах керування інфраструктурою ЕЦП на основі симетричної схеми розподілу ключів:

- Захист секретних ключів ЕЦП, що розподіляються централізовано.
- Захист сертифікатів абонентів мережі.
- Захист транспортного рівня системи, що забезпечує роботу процедур запитів і одержання сертифікатів, доставки інших файлів інфраструктури ЕЦП.

Заявка від сервісу на отримання ключа та сертифіката



Видача ключа та сертифіката сервісу корпоративної мережі

Рисунок 3.2 – Функціональна схема центру розподілу ключів

– Генерацію паролів для захисту секретних ключів від несанкціонованого доступу. Тип пароля може бути – випадковий (пароль буде створений випадковим образом з різних слів, що утворять випадкову фразу, що запам'ятовується легко), власний (можна задати за бажанням, але не менш 5 символів), випадковий цифровий (сформується випадковим образом з різних цифр).

– Формування інших ключів, що забезпечують відновлення симетричної ключової інформації й роботу іншого ПЗ.

Первісний набір симетричних ключів видається користувачеві в складі файлу ключового дистрибутива, зашифрованого на паролі, і який користувач одержує при його первинній реєстрації. До складу ключового дистрибутива входить персональний ключ зв'язку з УКЦ, ключ зв'язку зі своїм координатором, первинний секретний ключ підпису й сертифікат, сертифікати головних абонентів ЦРК. Інша ключова інформація надходить на комп'ютер після інсталяції ПЗ і в процесі відновлень.

Центр керування мережею

Центра керування забезпечує:

- Реєстрацію вузлів і абонентів корпоративної мережі, реєстрацію "Центрів реєстрації" зовнішніх користувачів.

- Взаємодія зі ЦРК і користувачами при керуванні сертифікатами.

- Формування захищених довідників доступу для вузлів мережі й довідників зв'язків вузлів і абонентів для УКЦ при штатній експлуатації й компрометації ключів абонентів.

- Інші функції.

Взаємодія абонентів зі ЦРК виробляється тільки через Центр керування мережею.

Центр реєстрації

Центр Реєстрації призначений для реєстрації зовнішніх користувачів і одержання для них цифрових сертифікатів. У Центрі Реєстрації при пред'явленні документів зовнішнім користувачем, що підтверджує його повноваження, створюється запит на сертифікат, виробляється відправлення його в ЦРК і здійснюється запровадження в дію виданого в ЦРК сертифіката. У Центрі Сертифікації сертифікат буде або задоволений, або відхилений. Тільки запит на сертифікат зі статусом "задоволений" стає сертифікатом підпису, і цей сертифікат може бути уведений у дію. Після введення в дію сертифіката, зовнішній користувач зможе користуватися ним (підписувати документи) на будь-якому вузлі із установленим ПЗ.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Центр Реєстрації виконує наступні функції:

- Генерація секретного ключа підпису й збереження його на персональному ключовому носії зовнішнього користувача.
- Уведення персональних даних для сертифіката зовнішнього користувача.
- Формування запиту на сертифікат.
- Підпис запиту на сертифікат ключем діючого адміністратора ЦР.
- Відправлення завіреного запиту в ЦРК
- Прийом сертифікатів зі ЦРК (відбувається автоматично).
- Перегляд запитів і прийнятих сертифікатів.
- Запровадження в дію сертифіката (збереження на персональному ключовому носії зовнішнього користувача).
- Формування запиту на відкликання сертифіката.
- Формування запиту на призупинення сертифіката.
- Формування запиту на поновлення сертифіката.

Крім того, Центр Реєстрації виконує експорт сертифікатів у різних кодуваннях.

Секретний ключ зовнішнього користувача і його сертифікат заносяться на його персональний носій. Це може бути дискета, компактний диск (CD), E-Token, смарт-карта, Touch memoгу і інші.

Секретний ключ зашифровується на паролі, вироблюваному програмою. Тип пароля може бути один з наступних:

- Власний пароль.
- Випадкова фраза, що запам'ятовується легко.
- Власний цифровий пароль.
- Випадковий цифровий пароль.

Розподіл відкритих ключів

Розглянемо основні алгоритми розподілу ключів.

На сьогоднішній день відомі наступні методи розподілу відкритих ключів:

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

- індивідуальне публічне оголошення відкритих ключів сервісами;
- використання привселюдно доступного каталогу відкритих ключів;
- участь авторитетного джерела відкритих ключів;
- сертифікати відкритих ключів.

Розглянемо кожний з перерахованих методів.

При *індивідуальному публічному оголошенні відкритих ключів* будь-яка сторона, що бере участь в обміні повідомленнями (X), може надати свій відкритий ключ (K_o) будь-якій іншій стороні. Недоліком даного підходу є неможливість забезпечити автентифікацію відправника відкритого ключа (K_o). Тобто, при даному підході в порушника з'являється можливість фальсифікації сервісів.

Використання привселюдно доступного каталогу відкритих ключів дозволяє домогтися більше високого ступеня захисту інформації й мережі побудованої з використанням сервіс-компонентної архітектури. У цьому випадку за ведення й поширення публічного каталогу повинна відповідати надійна організація (уповноважений об'єкт). При цьому повинні дотримуватися наступні правила.

1. Сервіси повинні реєструвати свої відкриті ключі в публічному каталозі, що веде вповноважений об'єкт.
2. Реєстрація повинна проходити по заздалегідь захищених каналах зв'язку.
3. Уповноважений об'єкт повинен періодично публікувати каталог відкритих ключів.

Недоліком даного підходу є наступне. Якщо порушникові вдасться змінити записи, що зберігаються в каталозі відкритих ключів, то він зможе авторитетно видавати фальсифіковані відкриті ключі й, отже, виступати від імені кожного з учасників обміну даними й читати повідомлення, призначені будь-якому сервісу.

Участь авторитетного джерела відкритих ключів. Обов'язковою умовою даного варіанта розподілу відкритих ключів сервісів є умова, що авторитетне джерело відкритих ключів має свій секретний ключ, і кожний сервіс знає його відкритий ключ. При цьому виконується наступний порядок дій:

1. Сервіс 1 надсилає запит авторитетному джерелу відкритих ключів про поточне значення відкритого ключа сервісу 2. При цьому вказується дата й час запиту (д.вр.).

2. Авторитетне джерело, використовуючи свій секретний ключ K_C^A , шифрує й передає повідомлення сервісу 1 $Y_1 = E_{K_C^A}(K_O^{П2}, \text{д.вр.})$, у якому втримується наступна інформація:

- $K_O^{П2}$ – відкритий ключ сервісу 2;
- буд. вр. – дата й час відправлення повідомлення.

3. Сервіс 1, використовуючи $K_O^{П2}$, шифрує й передає сервісу 2 шифроване повідомлення $Y_2 = E_{K_O^{П2}}(ID_1, N_1)$, що містить:

- ID_1 – ідентифікатор відправника (сервіс 1);
- N_1 – унікальну мітку даного повідомлення.

4, 5. Сервіс 2, одержавши шифроване повідомлення $Y_2 = E_{K_O^{П2}}(ID_1, N_1)$, дешифрує його за допомогою свого секретного ключа $K_C^{П2}$ $(ID_1, N_1) = D_{K_C^{П2}}(Y_2)$ й відповідно до ідентифікатора ID_1 , аналогічно з пунктами 1 і 2 вище перерахованих дій одержує від авторитетного джерела відкритий ключ сервісу 1 $K_O^{П1}$.

6. Сервіс 2, використовуючи $K_O^{П1}$, посилає сервісу 1 шифроване повідомлення $Y_4 = E_{K_O^{П1}}(N_1, N_2)$, де N_2 – унікальна мітка даного повідомлення.

7. Сервіс 1 шифрує за допомогою відкритого ключа $K_O^{П2}$ повідомлення Y , призначене сервісу 1 і передає $Y_5 = E_{K_O^{П2}}(Y, N_2)$.

Наведений варіант розподілу відкритих ключів має деякі недоліки:

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

– щораз, коли сервіс має намір передати інформацію новому адресатові, то він повинен звертатися до авторитетного джерела з метою одержання відкритого ключа;

– каталог імен і відкритих ключів, підтримуваний авторитетним джерелом, є привабливим місцем для порушника передачі інформації сервісів.

Сценарій розподілу відкритих ключів із застосуванням сертифікатів відкритих ключів. Обов'язковою умовою даного варіанта розподілу відкритих ключів сервісів є умова, що авторитетне джерело сертифікатів має свій секретний ключ K_C^A , і кожний сервіс знає його відкритий ключ K_O^A . При цьому виконується наступний порядок дій:

1. Сервіс 1 генерує пари ключів $[K_O^{П1}, K_C^{П1}]$ (відповідно, відкритий і секретний) і по захищеному каналу зв'язку звертається до авторитетного джерела сертифікатів з метою одержання сертифіката.

2. Авторитетне джерело шифрує за допомогою свого секретного ключа K_C^A сертифікат $C_{П1} = E_{K_C^A}^{A}[K_O^{П1}, ID_{П1}, T_{П1}]$ і видає його сервісу 1. Сертифікат містить:

– $K_O^{П1}$ – відкритий ключ сервісу 1 (даний ключ сервіс 1 сам згенерував і передав авторитетному джерелу для сертифікації);

– $ID_{П1}$ – ідентифікатор сервісу 1;

– $T_{П1}$ – термін дії сертифіката сервісу.

3. Сервіс 1 пересилає свій сертифікат $C_{П1} = E_{K_C^A}^{A}[K_O^{П1}, ID_{П1}, T_{П1}]$, отриманий від авторитетного джерела, сервісу 2. Останній, знаючи відкритий ключ авторитетного джерела сертифікатів K_O^A , має можливість прочитати й упевнитися, що отримане повідомлення є сертифікатом $D_{K_O^A}^{A}[C_{П1}] = D_{K_O^A}^{A}[E_{K_C^A}^{A}[K_O^{П1}, ID_{П1}, T_{П1}]] = [K_O^{П1}, ID_{П1}, T_{П1}]$.

4. 5, 6. Сервіс 2 виконує аналогічні дії, які були виконані сервісом 1 у пунктах 1, 2 і 3. Тобто одержує від авторитетного джерела сертифікат $C_{П2} = E_{K_C^A}^{A}[K_O^{П2}, ID_{П2}, T_{П2}]$. Пересилає його сервісу 1. Останній, знаючи відкритий

ключ авторитетного джерела сертифікатів K_O^A , має можливість прочитати й упевнитися, що отримане повідомлення є сертифікатом

$$D_{K_O^A} [C_{П2}] = D_{K_O^A} [E_{K_C^A} [K_O^{П2}, ID_{П2}, T_{П2}]] = [K_O^{П2}, ID_{П2}, T_{П2}].$$

У результаті перерахованих дій сервіси обмінялися відкритими ключами й готові до передачі й прийому сервісних повідомлень.

Застосування криптосистеми з відкритим ключем для розподілу секретних ключів

На сьогоднішній день існує кілька підходів застосування криптосистеми з відкритим ключем для розподілу секретних ключів [7]. Розглянемо деякі з них.

Простий розподіл секретних ключів складається у виконанні наступних дій:

1. Сервіс 1 генерує пари ключів $[K_O^{П1}, K_C^{П1}]$, відповідно, відкритий і секретний.
2. Сервіс 1 передає сервісу 2 повідомлення $X_{П1} = [ID_{П1}, K_O^{П1}]$, де $ID_{П1}$ – ідентифікатор сервісу 1.
3. Сервіс 2, одержавши повідомлення $X_{П1} = [ID_{П1}, K_O^{П1}]$ від сервісу 1, так само генерує свою пару ключів $[K_O^{П2}, K_C^{П2}]$.
4. Сервіс 2, використовуючи відкритий ключ $K_O^{П1}$ сервісу 1, шифрує й передає повідомлення $Y_{П2} = E_{K_O^{П1}} [ID_{П2}, K_C^{П2}]$ сервісу 1.
5. Сервіс 1 знищує свій секретний ключ $K_C^{П1}$, а сервіс 2 знищує відкритий ключ сервісу 1 $K_O^{П1}$.

Таким чином, обидва сервіси мають сеансовий (секретний) ключ $K_C^{П2}$ і можуть використовувати його для передачі інформації, захищеної традиційним шифруванням. По закінченні сеансу передачі інформації ключ $K_C^{П2}$ знищується. Однак даний підхід уразливий для активних порушень. Дійсно, якщо порушник має можливість впровадження в з'єднання між сервісами, те,

виконуючи наступні дії, він буде мати можливість знати секретний (сеансовий) ключ.

1. Сервіс 1 генерує пари ключів $[K_O^{\Pi 1}, K_C^{\Pi 1}]$ і передає сервісу 2 повідомлення $X_{\Pi 1} = [ID_{\Pi 1}, K_O^{\Pi 1}]$.

2. Порушник перехоплює повідомлення $X_{\Pi 1} = [ID_{\Pi 1}, K_O^{\Pi 1}]$, створює власну пару ключів $[K_O^H, K_C^H]$ і передає сервісу 2 повідомлення $X_H = [ID_{\Pi 1}, K_O^H]$.

3. Сервіс 2, одержавши повідомлення $X_H = [ID_{\Pi 1}, K_O^H]$, генерує свою пару ключів $[K_O^{\Pi 2}, K_C^{\Pi 2}]$, шифрує (використовуючи відкритий ключ порушника K_O^H) і передає повідомлення $Y_{\Pi 2} = E_{K_O^H}[ID_{\Pi 2}, K_C^{\Pi 2}]$ сервісу 1.

4. Порушник перехоплює повідомлення $Y_{\Pi 2} = E_{K_O^H}[ID_{\Pi 2}, K_C^{\Pi 2}]$, дешифрує його $[ID_{\Pi 2}, K_C^{\Pi 2}] = D_{K_C^H}[Y_{\Pi 2}]$, визначає сеансовий ключ $K_C^{\Pi 2}$ і передає сервісу 2 повідомлення $Y_{\Pi 2} = E_{K_O^{\Pi 1}}[ID_{\Pi 2}, K_C^{\Pi 2}]$.

У результаті обидва сервіси мають сеансовий ключ $K_C^{\Pi 2}$, однак не будуть підозрювати, що він теж відомий і порушникові.

Сценарій розподілу секретних ключів із забезпеченням конфіденційності й автентичності і складається у виконанні наступних дій.

1. Сервіси генерують пари ключів, відповідно $[K_O^{\Pi 1}, K_C^{\Pi 1}]$, $[K_O^{\Pi 2}, K_C^{\Pi 2}]$, і обмінюються між собою відкритими ключами $K_O^{\Pi 1}$ й $K_O^{\Pi 2}$.

2. Сервіс 1, використовуючи $K_O^{\Pi 2}$, передає сервісу 2 повідомлення $Y_{\Pi 1} = E_{K_O^{\Pi 2}}[ID_{\Pi 1}, N_1]$, що містить: свій ідентифікатор – $ID_{\Pi 1}$; N_1 – унікальна мітка даних повідомлень.

3. Сервіс 2, використовуючи $K_O^{\Pi 1}$, передає сервісу 1 повідомлення $Y_{\Pi 2} = E_{K_O^{\Pi 1}}[N_2, N_1]$, що містить N_1 і N_2 – унікальні мітки даного

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

повідомлення. Наявність мітки N_1 переконує сервісу 1 у тім, що тільки сервіс 2 міг дешифрувати повідомлення $Y_{\Pi} = E_{K_O^{\Pi_2}}[ID_{\Pi}, N_1]$.

4. Сервіс 1, використовуючи $K_O^{\Pi_2}$, передає сервісу 2 повідомлення $Y_{\Pi} = E_{K_O^{\Pi_2}}[N_2]$, що містить унікальну мітку N_2 . Дане повідомлення виконує функцію підтвердження для сервісу 2, що його респондентом є сервіс 1.

5. Сервіс 1 генерує секретний (сеансовий) ключ K_C , що двічі шифрується з використанням: свого секретного ключа $E_{K_C^{\Pi_1}}[K_C]$ й відкритого ключа сервісу 2 $E_{K_O^{\Pi_2}}[E_{K_C^{\Pi_1}}[K_C]]$. Після виконання процедури шифрування повідомлення $Y_{\Pi} = E_{K_O^{\Pi_2}}[E_{K_C^{\Pi_1}}[K_C]]$ передається сервісу 2. Останній, маючи відкритий ключ сервісу 1 і свій секретний ключ, дешифрує отримане повідомлення.

У результаті перераховані дії обидва сервіси мають секретний (сеансовий) ключ K_C .

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:

- Інтерфейс ПЗ.
- Припинення строку дії
- Зміна статусу сертифікату.
- Використання сертифіката.
- Зберігання сертифіката.
- Реєстрація/видання сертифіката.
- Розповсюдження сертифіката.
- Генерація пари ключів.
- Транспортування ключа.

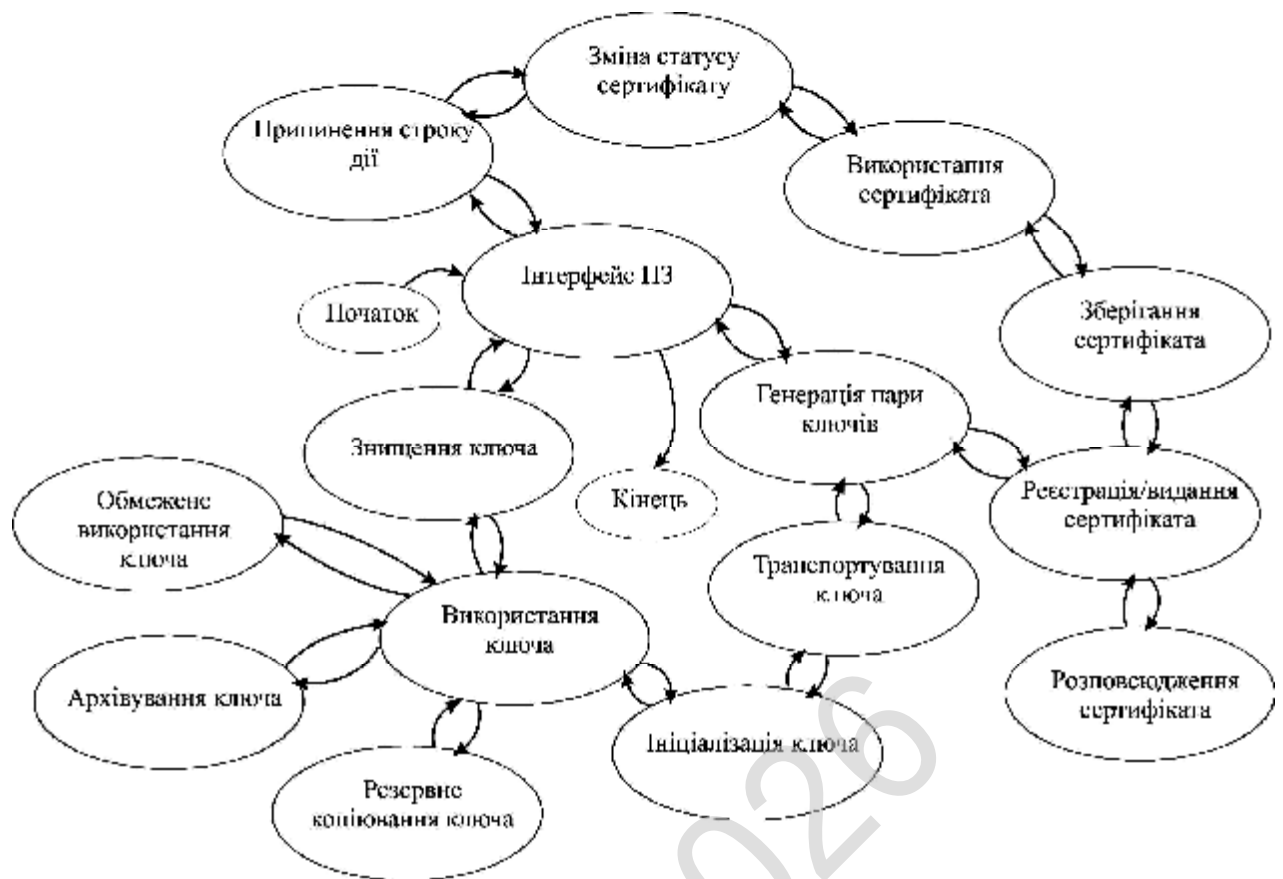


Рисунок 3.3 – Діаграма взаємодії процесів

- Ініціалізація ключа.
- Знищення ключа.
- Використання ключа.
- Резервне копіювання ключа.
- Архівування ключа.
- Обмежене використання ключа.

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 РОБОТИ. 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.

При розробці використовувались концепції діаграм діяльності. Тобто в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Це фундаментальна одиниця визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії.

Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності.

Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

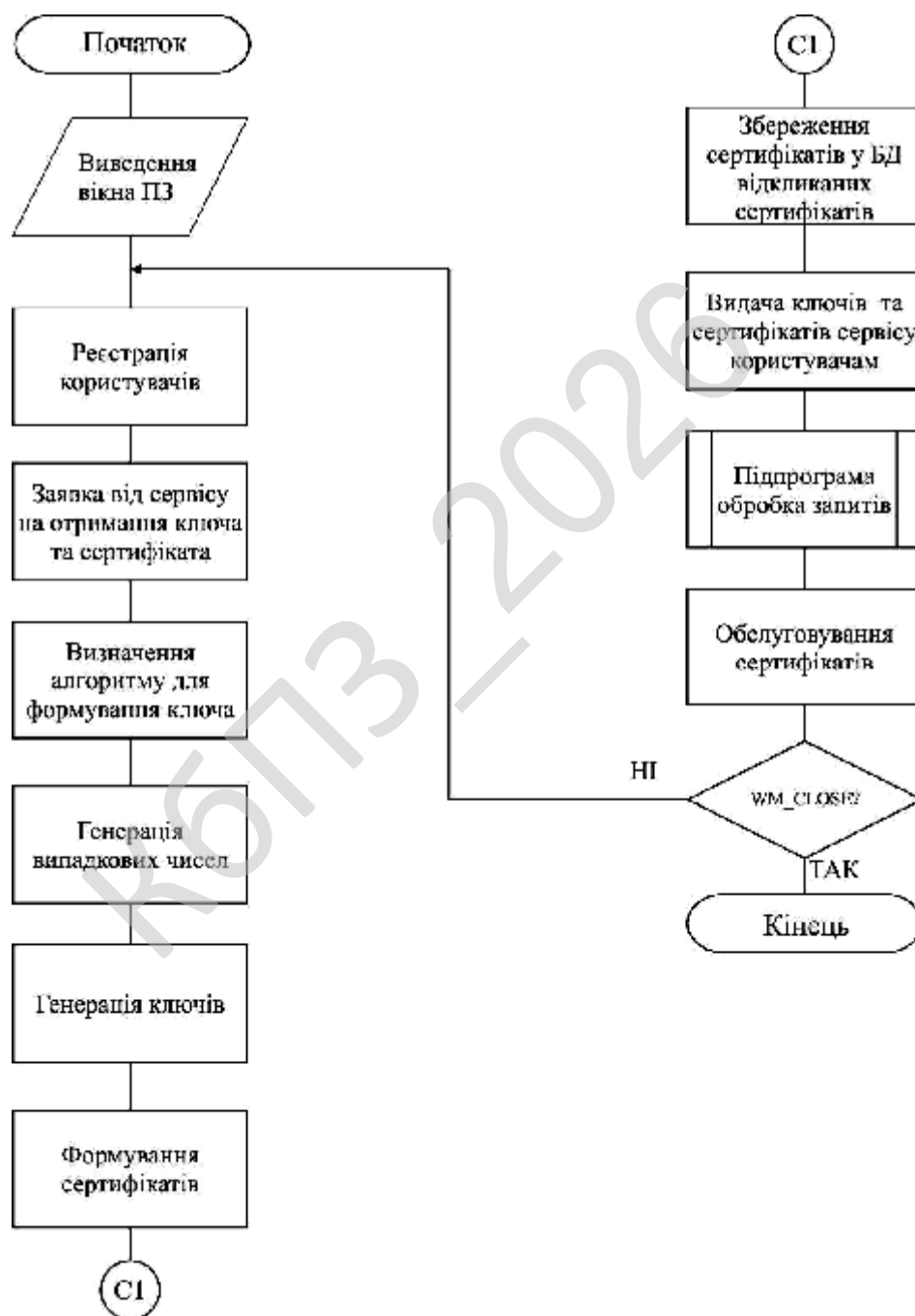


Рисунок 4.1 – Блок схема основної програми

Нижче приведемо частину коду основної програми, яка реалізує ці алгоритми.

У якості алгоритмів шифрування даних у мережі, для яких здійснюється розподіл ключів, обрані алгоритми RSA та DSA.

```
//створення ключів RSA
private static void Create_RSAKeys()
{
    CspParameters csp = new CspParameters();
    csp.KeyContainerName = "RSA Тест";
    const int PROV_RSA_FULL = 1;
    csp.ProviderType = PROV_RSA_FULL;
    const int AT_KEYEXCHANGE = 1;
    // const int AT_SIGNATURE = 2;
    csp.KeyNumber = AT_KEYEXCHANGE;
    _RSACryptoServiceProvider rsa =
        new _RSACryptoServiceProvider(1024, csp);
    rsa.PersistKeyInCsp = false;
```

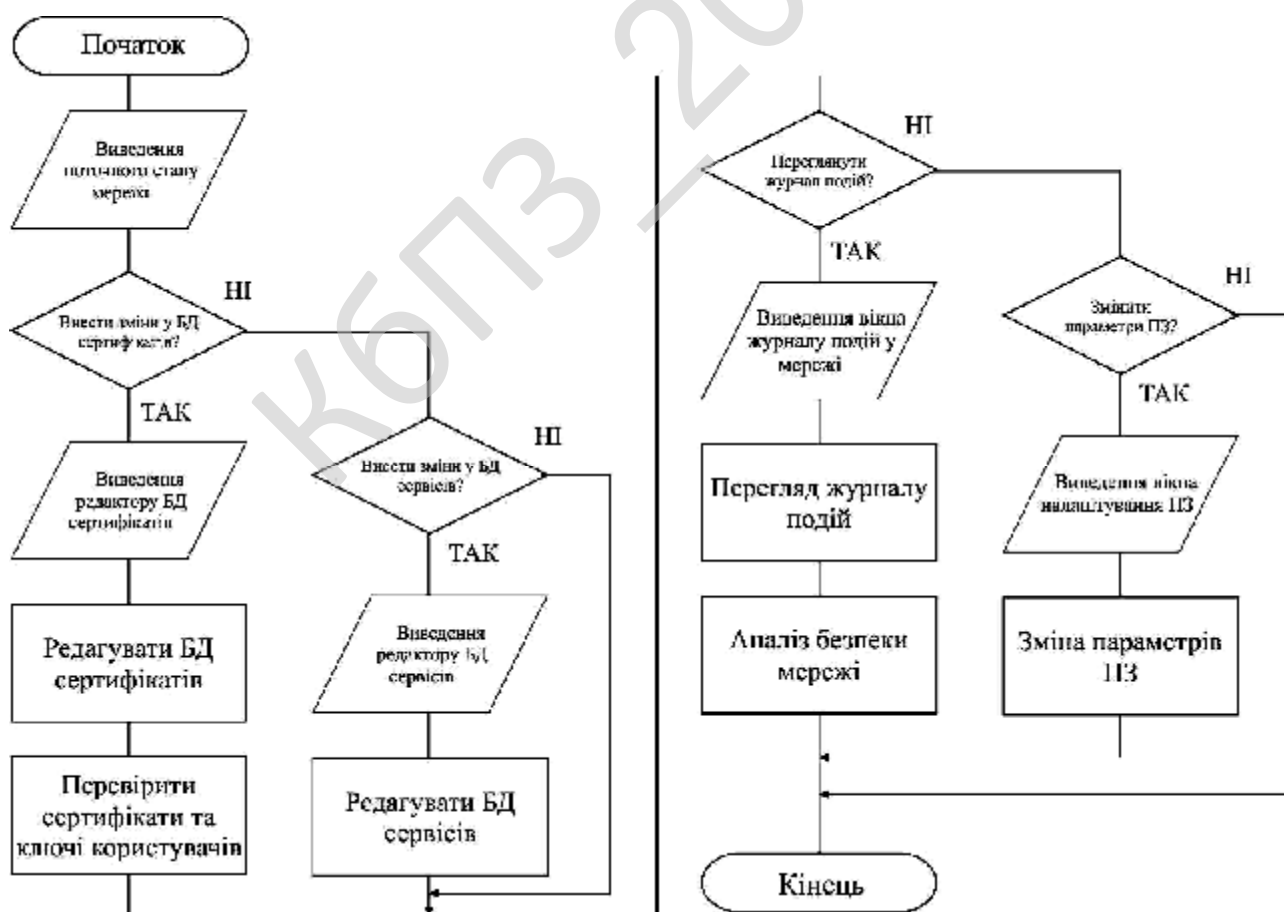


Рисунок 4.2 – Блок схема підпрограми

```

// Ключ шифрування
    AsnKeyBuilder.AsnMessage key = null;

// Таємний ключ
    _RSAParameters privateKey = rsa.ExportParameters(true);
    key = AsnKeyBuilder.PrivateKeyToPKCS8(privateKey);
    using (BinaryWriter writer = new BinaryWriter(
        new FileStream("private.rsa.cs.ber", FileMode.Create,
            FileAccess.ReadWrite)))
    {
        writer.Write(key.GetBytes());
    }

//тестування ключів DSA
    private static void TestDsaKeys()
    {
        CreateDsaKeys();
        LoadDsaPrivateKey();
        LoadDsaPublicKey();
    }

//тестування ключів RSA
    private static void Test_RSAKeys()
    {
        Create_RSAKeys();
        Load_RSAPrivateKey();
        Load_RSAPublicKey();
    }

// Відкритий ключ
    _RSAParameters publicKey = rsa.ExportParameters(false);
    key = AsnKeyBuilder.PublicKeyToX509(publicKey);
    using (BinaryWriter writer = new BinaryWriter(
        new FileStream("public.rsa.cs.ber", FileMode.Create,
            FileAccess.ReadWrite)))
    {
        writer.Write(key.GetBytes());
    }

    rsa.Clear();
}

// Створення ключів DSA
    private static void CreateDsaKeys()
    {
        CspParameters csp = new CspParameters();
        csp.KeyContainerName = "DSA Тест";
    }

```

```

        const int PROV_DSS_DH = 13;
        csp.ProviderType = PROV_DSS_DH;
// Не використовуйте AT_EXCHANGE для створення. Це алгоритм підпису
        const int AT_SIGNATURE = 2;
        csp.KeyNumber = AT_SIGNATURE;
        DSACryptoServiceProvider dsa =
            new DSACryptoServiceProvider(1024, csp);
        dsa.PersistKeyInCsp = false;
// Ключ шифрування
        AsnKeyBuilder.AsnMessage key = null;
// Таємний ключ
        DSAParameters privateKey = dsa.ExportParameters(true);
        key = AsnKeyBuilder.PrivateKeyToPKCS8(privateKey);
        using (BinaryWriter writer = new BinaryWriter(
            new FileStream("private.dsa.cs.ber", FileMode.Create,
                FileAccess.ReadWrite)))
        {
            writer.Write(key.GetBytes());
        }
// Редагування таємного ключа RSA
        private static void Load_RSAPrivateKey()
        {
//
// Редагування Таємного ключа
//   PKCS#8 формат
//
            AsnKeyParser keyParser =
                new AsnKeyParser("private.rsa.cs.ber");
            _RSAParameters privateKey = keyParser.Parse_RSAPrivateKey();
//
// Ініціалізація CSP
//   Подавляє створення нового ключа
//
            CspParameters csp = new CspParameters();
            csp.KeyContainerName = "RSA Тест";
            const int PROV_RSA_FULL = 1;
            csp.ProviderType = PROV_RSA_FULL;
            const int AT_KEYEXCHANGE = 1;
// const int AT_SIGNATURE = 2;
            csp.KeyNumber = AT_KEYEXCHANGE;
//
// Ініціалізація криптопровайдер

```

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

```

//
    _RSACryptoServiceProvider rsa =
        new _RSACryptoServiceProvider(csp);
    rsa.PersistKeyInCsp = false;
//
//
//
    rsa.ImportParameters(privateKey);

    rsa.Clear();
}
// Відкритий ключ
    DSAParameters publicKey = dsa.ExportParameters(false);
    key = AsnKeyBuilder.PublicKeyToX509(publicKey);
    using (BinaryWriter writer = new BinaryWriter(
        new FileStream("public.dsa.cs.ber", FileMode.Create,
            FileAccess.ReadWrite)))
    {
        writer.Write(key.GetBytes());
    }

    dsa.Clear();
}
// Редагування таємного ключа DSA
private static void LoadDsaPrivateKey()
{
    // Редагування таємного ключа
    // PKCS#8 формат
    AsnKeyParser keyParser =
        new AsnKeyParser("private.dsa.cs.ber");
    DSAParameters privateKey = keyParser.ParseDSAPrivateKey();
// Ініціалізація CSP
// Подавляє створення нового ключа
    CspParameters csp = new CspParameters();
    csp.KeyContainerName = "DSA Тест";
// Не використовуйте PROV_DSS_DH для редагування.
// const int PROV_DSS_DH = 13;
    const int PROV_DSS = 3;
    csp.ProviderType = PROV_DSS;
// const int AT_EXCHANGE = 1;
    const int AT_SIGNATURE = 2;
    csp.KeyNumber = AT_SIGNATURE;

```

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

```

//
// Ініціалізація криптопровайдера
//
    DSACryptoServiceProvider dsa =
        new DSACryptoServiceProvider(csp);
    dsa.PersistKeyInCsp = false;
//
//
    dsa.ImportParameters(privateKey);
    dsa.Clear();
}
// Редагування відкритого ключа DSA
private static void LoadDsaPublicKey()
{
//
// Редагування відкритого ключа
// X.509 формат
//
    AsnKeyParser keyParser =
        new AsnKeyParser("public.dsa.cs.ber");
    DSAPublicKey publicKey = keyParser.ParseDSAPublicKey();
//
// Ініціалізація CSP
// Подавляє створення нового ключа
//
    CspParameters csp = new CspParameters();
// const int PROV_DSS_DH = 13;
    const int PROV_DSS = 3;
    csp.ProviderType = PROV_DSS;
    const int AT_SIGNATURE = 2;
    csp.KeyNumber = AT_SIGNATURE;
    csp.KeyContainerName = "DSA Test (OK to Delete)";
//
// Ініціалізація криптопровайдера
//
    DSACryptoServiceProvider dsa =
        new DSACryptoServiceProvider(csp);
    dsa.PersistKeyInCsp = false;
//
//
//
    dsa.ImportParameters(publicKey);

```

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

```

        dsa.Clear();
    }
// Редагування відкритого ключа RSA
    private static void Load_RSAPublicKey()
    {
//
// Редагування Відкритого ключа
// X.509 формат
//
        AsnKeyParser keyParser =
            new AsnKeyParser("public.rsa.cs.ber");
        _RSAParameters publicKey = keyParser.Parse_RSAPublicKey();
//
// Ініціалізація CSP
// Подавляє створення нового ключа
//
        CspParameters csp = new CspParameters();
        csp.KeyContainerName = "RSA Тест";
        const int PROV_RSA_FULL = 1;
        csp.ProviderType = PROV_RSA_FULL;
        const int AT_KEYEXCHANGE = 1;
// const int AT_SIGNATURE = 2;
        csp.KeyNumber = AT_KEYEXCHANGE;
//
// Ініціалізація криптопровайдера
//
        _RSACryptoServiceProvider rsa =
            new _RSACryptoServiceProvider(csp);
        rsa.PersistKeyInCsp = false;
//
        rsa.ImportParameters(publicKey);
        rsa.Clear();
    }
}

```

4.2 Захист розробленого програмного забезпечення

Дані в програмі захищаються за допомогою використання алгоритму Md5. Він отримує на вході повідомлення довільної довжини і створює на виході

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

дайджест повідомлення довжиною 128 біт. Алгоритм складається з наступних кроків:

1. Додавання недостаючих біт. Повідомлення доповнюється так, щоб його довжина стала рівна 448 по модулю 512 (довжина $448 \bmod 512$). Це означає, що довжина доданого повідомлення на 64 біта менше, ніж число, кратне 512. Додавання проводиться завжди, навіть якщо повідомлення має потрібну довжину. Наприклад, якщо довжина повідомлення 448 біт, воно доповнюється 512 бітами до 960 біт. Таким чином, число біт, що додаються, знаходиться в діапазоні від 1 до 512.

Додавання складається з одиниці, за якою слідує необхідна кількість нулів.

2. Додавання довжини. 64-бітове представлення довжини початкового (до додавання) повідомлення в бітах приєднується до результату першого кроку. Якщо первинна довжина більша, ніж 264, то використовуються тільки останні 64 біта. Таким чином, поле містить довжину початкового повідомлення по модулю 264.

В результаті перших двох кроків створюється повідомлення, довжина якого кратна 512 бітам. Це розширене повідомлення представляється як послідовність 512-бітових блоків $Y_0, Y_1, \dots, Y_{l-1}$, при цьому загальна довжина розширеного повідомлення рівна $L \cdot 512$ бітам. Таким чином, довжина отриманого розширеного повідомлення кратна шістнадцяти 32-бітовим словам.

3. Ініціалізація MD-буфера. У алгоритмі Md5 використовується 128-бітовий буфер для зберігання проміжних і остаточних результатів хеш-функції. Буфер може бути представлений як чотири 32-бітові регістри (A, B, C, D). Ці регістри ініціалізувалися наступними шістнадцятковими числами:

$$A = 01234567$$

$$B = 89abcdef$$

$$C = Fedcba98$$

$$D = 76543210$$

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

4. Обробка послідовності 512-бітових (16-словних) блоків. Основою алгоритму Md5 є модуль, що складається з чотирьох циклічних обробок, позначений як Hmd5. Чотири цикли мають схожу структуру, але кожен цикл використовує свою елементарну логічну функцію, ff , що позначається, fg , fh і fi відповідно.

Кожен цикл приймає на вхід поточний 512-бітовий блок Y_q , що обробляється в даний момент, і 128-бітове значення буфера ABCD, яке є проміжним значенням дайджесту, і змінює вміст цього буфера. Кожен цикл також використовує четверту частину 64-елементної таблиці $T[1 \dots 64]$, побудованої на основі функції $\sin$. i -ий елемент T , $T[i]$, що позначається, має значення, рівне цілій частині від $232 * \text{abs}(\sin(i))$, і задане в радіанах. Оскільки $\text{abs}(\sin(i))$ є числом між 0 і 1, кожен елемент T є цілим, яке може бути представлене 32 бітами. Таблиця забезпечує “випадковий” набір 32-бітових значень, які повинні ліквідовувати будь-яку регулярність у вхідних даних.

Для отримання Md_{q+1} вихід чотирьох циклів складається по модулю 232 з Md_q . Складання виконується незалежно для кожного з чотирьох слів в буфері.

5) Вихід Md5. Після обробки всіх L 512-бітових блоків виходом L -ої стадії є 128-бітовий дайджест повідомлення.

Детальніше логіку кожного з чотирьох циклів виконання одного 512-бітового блоку розглянуто нижче. Кожен цикл складається з 16 кроків, що оперують з буфером ABCD.

$$A \leftarrow B + \text{Clss}(A + f(B, C, D) + X[k] + T[i]),$$

A, B, C, D – чотири слова буфера; після виконання кожного окремого кроку відбувається циклічне зрушення вліво на одне слово.

f – одна з елементарних функцій ff, fg, fh, fi .

CLSs – циклічне зрушення вліво на s біт 32-бітового аргументу.

$X[k] = M[q * 16 + k]$ – кодує 32-бітове слово в q -ому 512 блоці повідомлення.

$T[i]$ – i -е 32-бітове слово в матриці T .

+ – складання по модулю 232.

На кожному з чотирьох циклів алгоритму використовується одна з чотирьох елементарних логічних функцій. Кожна елементарна функція отримує три 32-бітові слова на вході і на виході створює одне 32-бітове слово. Кожна функція є безліччю побітових логічних операцій, тобто n -ий біт виходу є функцією від n -ого біта трьох входів. Елементарні функції наступні:

$$ff = (B \& C) \text{ (not } B \& D)$$

$$fg = (B \& D) \vee (C \& \text{not } D)$$

$$fh = B \ C \ D$$

$$fi = C \ (B \& \text{not } D)$$

Масив з 32-бітових слів $X[0..15]$ містить значення поточного 512-бітового вхідного блоку, який обробляється зараз. Кожен цикл виконується 16 разів, а оскільки кожен блок вхідного повідомлення обробляється в чотирьох циклах, то кожен блок вхідного повідомлення обробляється по схемі 64 рази. Якщо представити вхідний 512-бітовий блок у вигляді шістнадцяти 32-бітових слів, то кожне вхідне 32-бітове слово використовується чотири рази, по одному разу в кожному циклі, і кожен елемент таблиці T , що складається з 64 32-бітових слів, використовується тільки один раз. Після кожного кроку циклу відбувається циклічне зрушення вліво чотирьох слів A, B, C і D . На кожному кроці змінюється тільки одне з чотирьох слів буфера $ABCD$. Отже, кожне слово буфера змінюється 16 разів, і потім 17-й раз в кінці для отримання остаточного виходу даного блоку.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

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

- Меню.
- Функціональні кнопки.
- Ресурси.

У вищезазначених розділах реалізується наступний функціонал:

- З'єднання.
- Розподіл ключів.
- Сертифікати.
- Ресурси.
- Журнал.
- Параметри.
- Довідка.

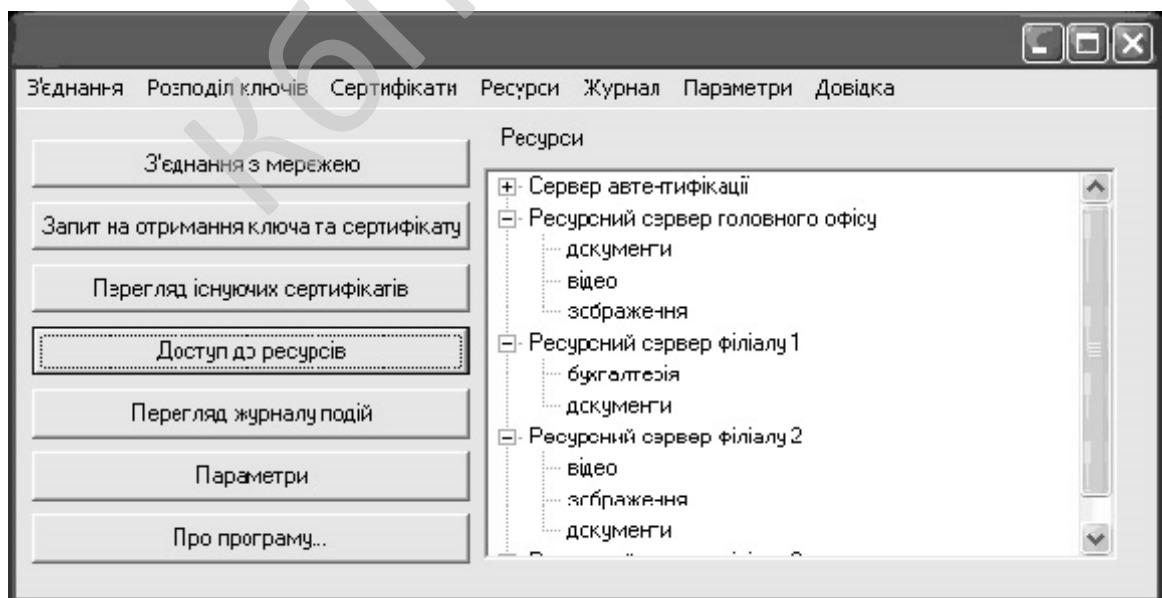


Рисунок 5.1– Основне вікно програми

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

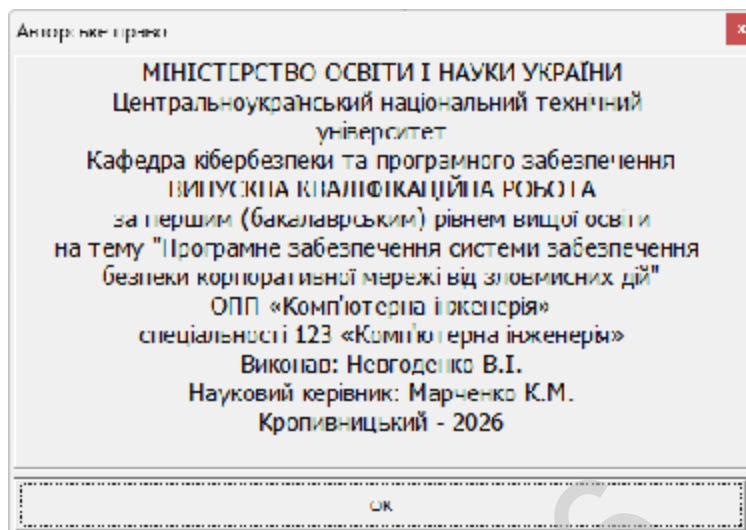


Рисунок 5.2 – Авторське право

Обрано умови розповсюдження – Shareware. Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми. В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання. Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareєstrуватися), заплативши авторіві певну суму.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи забезпечення безпеки корпоративної мережі від зловмисних дій.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем забезпечення безпеки корпоративної мережі від зловмисних дій.
- Досліджена система забезпечення безпеки корпоративної мережі від зловмисних дій.
- На основі отриманих результатів досліджень створена програмна реалізація системи забезпечення безпеки корпоративної мережі від зловмисних дій.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання забезпечення безпеки корпоративної мережі від зловмисних дій.

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

Програма реалізована на мові високого рівня Visual C#. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи забезпечення безпеки корпоративної мережі від зловмисних дій. Це

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

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 Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Md5.

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.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.
2. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.
3. Kuznetsov O., Frontoni E., Kuznetsova Y., Chevardin V., Smirnov O. «Architectural foundations for adaptive security in edge computing systems». *Cybersecurity Defensive Walls in Edge Computing*, 2025. pp. 21-61.
4. 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.
5. 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
6. 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.
7. 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.

8. 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.

9. 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.

10. 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.

11. 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.

12. 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.

13. 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.

14. Смірнова Т.В., Коноплицька-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.

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

15. 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.
16. 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.
17. 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.
18. Akhalaia, G., Iavich, M., Iashvili, G., Prysiashnyy, D., Smirnova, T. «Secure Encrypted Connection on Georgian Website». *CEUR Workshop Proceedings*, 2023, 3550, pp. 313-320.
19. 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
20. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
21. 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.
22. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Beбeshko, B. «Selection of a Rational Composition of Information 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.

23. 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.

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

25. 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.

26. 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.

27. 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.

28. 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,

29. 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

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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

35. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». *4 IEEE*

					ВКРБ-123.26.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

International Conference on Advanced Information and Communication Technologies (AICT) - 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

36. 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.*

37. 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.*

38. 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.*

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

40. 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.*

41. 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.*

42. 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.*

43. 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.*

44. 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.
45. 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.
46. 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.
47. 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.
48. 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.
49. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». *International Journal of Computer Network and Information Security (IJCNIS)*. Vol. 12, No. 3, 2020. PP.33-43.
50. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 646-660.

51. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.
52. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.
53. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, Vol 2588, P. 215-227, 2019.
54. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.
55. Smirnov, O., Kuznetsov, A., Kiian, A., Gorbenko, Y., Cherep, O., Bexhter L. «Code-based Pseudorandom Generator for the Post-Quantum Period», *2019 IEEE International Conference on Advanced Trends in Information Theory (IEEE ATIT 2019)*. 18.12.19-20.12.19 Kyiv Ukraine. P. 204 – 209.
56. Smirnov, O., Kuznetsov, A., Nariezhnii, O., Stelnyk, S., Kokhanovska, T., Kuznetsova T., «Side Channel Attack on a Quantum Random Number Generator», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18 - 21 September 2019. P.713-718.